

Installation and User's Guide



CORPORATE HEADQUARTERS: 275 Paterson Ave, Little Falls, NJ 07424 • (973) 890-7300 • Fax: (973) 890-7147
E-mail: support@fdrinnovation.com • sales@fdrinnovation.com • <http://www.innovationdp.fdr.com>

EUROPEAN OFFICES:	FRANCE 01-49-69-94-02	GERMANY 089-489-0210	NETHERLANDS 036-534-1660	UNITED KINGDOM 0208-905-1266	NORDIC COUNTRIES +31-36-534-1660
------------------------------	--------------------------	-------------------------	-----------------------------	---------------------------------	-------------------------------------

THIS PAGE WAS INTENTIONALLY LEFT BLANK

We at INNOVATION DATA PROCESSING have the pleasure of presenting to you the FDRSOS User Manual.

**PURPOSE OF
THE GUIDE**

The purpose of this guide is to provide you with the information to install, use and understand FDRSOS.

**WHAT IS
FDRSOS?**

FDRSOS provides disaster/recovery of Open System disk volumes by backing them up to and restoring from MVS-attached mainframe tapes and disks.

Open System Disk volumes are disks formatted for use by Unix systems and certain other systems such as OS/2®, Windows NT® and Novell®Netware®.

EMC Symmetrix® ESP-capable storage subsystems allow MVS systems to read and write Open System volumes, but only using the special I/O techniques built into FDRSOS.

Performance. FDRSOS supports ESCON channels for disk and tape access, and supports the newest high-speed, hi-capacity tape drives, to provide high-performance backup and restore..

FDRSOS is a trademark of INNOVATION DATA PROCESSING and is registered with the US patent office.

FDRSOS is a proprietary program product of INNOVATION DATA PROCESSING, Innovation Plaza, 275 Paterson Avenue, Little Falls, New Jersey 07424.
It is available for license exclusively from INNOVATION DATA PROCESSING, INC.

**EUROPEAN
OFFICES:**

France

Innovation Data Processing S.A.R.L.
66 Avenue François Arago
92000 Nanterre
Tel: 01-47-69-15-15 Fax: 01-47-69-15-16
E-mail: 76543.1377@compuserve.com

Germany

Innovation Data Processing
International Ltd.
Orleansstraße 4a
D-81669 München
Tel: 089-489 0210 Fax: 089-489 1355
E-mail: 76543.1401@compuserve.com

Netherlands

Innovation Data Processing
Brouwerstraat 8
1315 BP Almere
Tel: 036-534 1660 Fax: 036-533 7308
E-mail: 76543.1376@compuserve.com

United Kingdom

Innovation Data Processing Ltd.
Clarendon House
125 Shenley Road
Borehamwood, Herts, WD6 1AG
Tel: 0208-905 1266 Fax: 0208-905 1428
E-mail: 76543.1374@compuserve.com

© Copyright 1996, 1997

© INNOVATION DATA PROCESSING, INC.

Reproduction of this User Manual is prohibited except by licensed users for their internal use.

Main Phone Number (US) 973-890-7300 Fax Number (US) 973-890-7147 E-mail (US): 76322.2076@compuserve.com
<http://www.innovationdp.fdr.com>

THIS PAGE WAS INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

Page No.

FDRSOS INTRODUCTION AND OVERVIEW

	SUMMARY OF MODIFICATIONS V5.3 LEVEL03	v
	SUMMARY OF MODIFICATIONS V5.3 LEVEL02	v
200.01	FDRSOS OVERVIEW	200-1
200.02	FDRSOS CONCEPTS	200-2
200.03	FDRSOS IMPLEMENTATION	200-5
200.04	FDRSOS VOLUME STATUS DISPLAYS	200-11

FDRSOS JCL STATEMENTS EXAMPLES

210.01	FDRSOS JOB CONTROL REQUIREMENTS	210-1
210.02	FDRSOS DUMP STATEMENT	210-7
210.03	FDRSOS RESTORE STATEMENT	210-10
210.04	FDRSOS PRINT STATEMENT	210-13
210.05	FDRSOS ERASE STATEMENT	210-15
210.06	FDRSOS LABEL STATEMENT	210-17
210.07	FDRSOS VARYON/VARYOFF STATEMENT	210-18
210.08	FDRSOS MOUNT STATEMENT	210-20
210.09	FDRSOS SELECT/EXCLUDE STATEMENT	210-23
210.10	FDRSOS PRINT MODIFIERS	210-24
210.20	FDRSOS INITIALIZATION EXAMPLES	210-26
210.21	FDRSOS BACKUP EXAMPLES	210-29
210.22	FDRSOS RESTORE EXAMPLES	210-32
210.23	FDRSOS PRINT EXAMPLES	210-34
210.24	FDRSOS ERASE EXAMPLES	210-36

TABLE OF CONTENTS

Page No.

FDRSOS UTILITIES AND SPECIAL CONSIDERATIONS

220.01	FDRSOSTC BACKUP COPY UTILITY	220-1
220.02	FDRSOSTC JCL REQUIREMENTS	220-2
220.03	FDRSOSTC CONTROL STATEMENTS.....	220-3
220.04	FDRSOSTC EXAMPLES	220-4
220.05	FDRSOS OPEN SYSTEM UTILITIES	220-6
220.10	FDRSOS SPECIAL CONSIDERATIONS.....	220-7
220.11	FDRSOS COORDINATION	220-8
220.12	FDRSOS AND OTHER BACKUPS	220-11
220.13	FDRSOS AND FDR/UPSTREAM.....	220-12
220.14	FILE RESTORES FROM FDRSOS BACKUPS.....	220-14
220.15	FDRSOS DIAGNOSTIC TECHNIQUES.....	220-16

FDRSOS INSTALLATION AND MESSAGES

230.01	FDRSOS INSTALLATION	230-1
230.02	INSTALLING THE INSTALLATION CONTROL LIBRARY	230-2
230.03	INSTALLING THE PROGRAM LIBRARY.....	230-4
230.04	INSTALLING THE JCL LIBRARY.....	230-5
230.10	FDRZAPOP GLOBAL OPTION CHANGE FACILITY.....	230-6
230.11	FDRZAPOP STATEMENTS AND OPTIONS	230-7
230.20	FDRSOS MESSAGES	230-9
230.21	FDRSOS CONSOLE MESSAGES.....	230-10
230.22	FDRSOS PRINTER MESSAGES	230-11
230.23	FDRSOS ABEND CODES	230-25

SUMMARY OF MODIFICATIONS**Summary of Modifications for FDRSOS V5.3 LEVEL03****BCV
VOLUMES**

EMC Symmetrix supports Business Continuance Volumes (BCVs). BCVs are extra mirrors of volumes in the Symmetrix. These BCVs can be "split" from their primary volumes, creating an instant frozen copy of the volume. FDRSOS can then backup the frozen BCV, even though updates are occurring to the primary volume. When the backup is complete, the BCV can be reconnected to its primary volume, and any tracks that were updated while they were split are copied to the BCV.

This process allows you to quiesce your Open System volumes for a very short period of time, just long enough to split the BCVs of the volumes to be processed. The backup can then be done without time pressure.

V5.3 level 03 of FDRSOS contains beta test code for support of BCV volumes. It is not documented in this manual. Any user who wishes to test FDRSOS support for BCV volumes should contact Innovation for the necessary documentation. If you are not already using BCV mirrors of your Open System volumes, you will need to update your Symmetrix configuration to define the BCV volumes; if you have no unused disk space in the Symmetrix, you may need to add disks to the subsystem.

VOLRESET

Please note that the VOLRESET= parameter was incorrectly documented in the V5.3 level 02 manual. The meanings were exactly reversed, VOLRESET=YES should have been VOLRESET=NO and vice versa. However, in Appendix D (AIX/6000 restore), the use of VOLRESET was correct. The documentation in this manual is correct.

VARYPATH

A new FDRSOS command, VARYPATH, is available for use when one or more paths to an FDRSOS volume are offline. Normal MVS VARY commands cannot put those paths back online, but the VARYPATH command can do so, making all paths available if they are physically online.

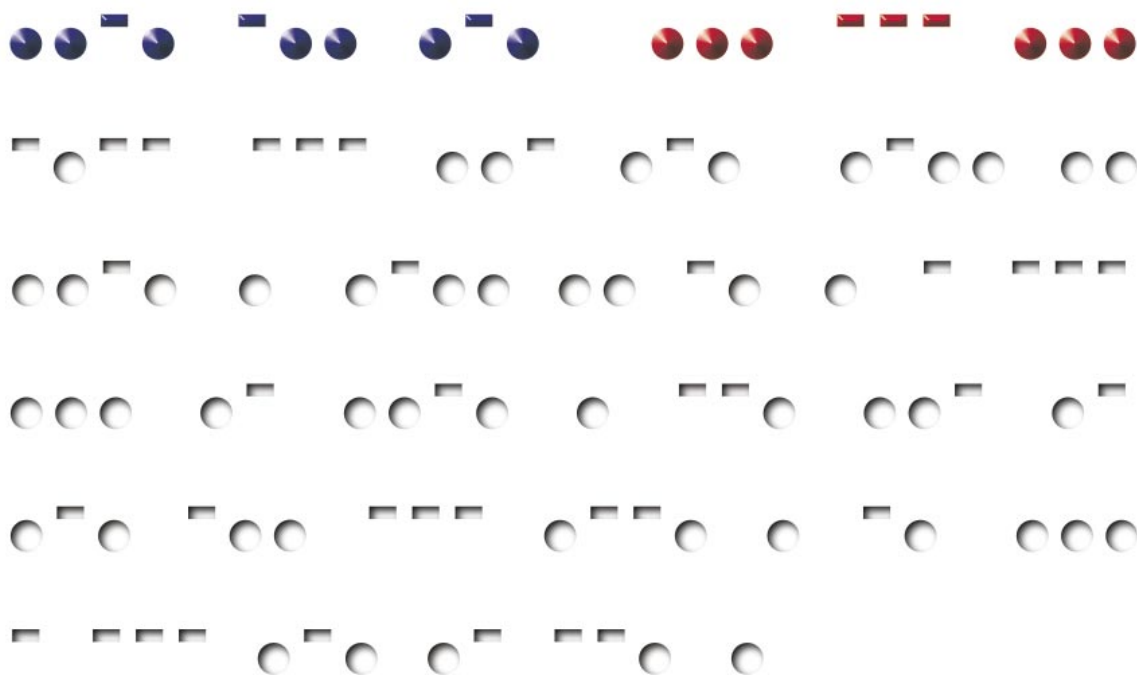
**HP/UX AND
SUN SOLARIS**

FDRSOS now contains enhanced support for HP/UX and SUN SOLARIS systems. PRINT=STATUS and VOLRESET=NO now support these systems.

Summary of Modifications for FDRSOS V5.3 LEVEL02

Version 5.3 level 02 was the initial production release of FDRSOS.

THIS PAGE WAS INTENTIONALLY LEFT BLANK



Installation and User's Guide

THIS PAGE WAS INTENTIONALLY LEFT BLANK

200.01 FDRSOS OVERVIEW

FDRSOS (Safeguard Open Storage) provides high-speed, reliable backup and restore of Open System (SCSI) data on an MVS system (including MVS/XA, MVS/ESA, and OS/390).

When used with an appropriately configured EMC Symmetrix Storage Subsystem, FDRSOS under MVS can be used to backup and restore disk volumes which are actually being used as SCSI disks by a UNIX system, a PC, or a server. Backups can be output to MVS tape, or to data sets on MVS DASD volumes.

FDRSOS uses the data management strength of IBM's S/390 hardware and MVS software data management to provide high-speed backups which are easily managed.

FDRSOS full-volume backups include all data sectors on the selected disks. A facility to backup and restore user-specified data sectors from Open System disks is provided.

FDRSOS also offers a function to print data from Open System disks and to scan for specific ASCII or hex data strings.

EMC SYMMETRIX

FDRSOS is possible because of a teaming agreement between Innovation Data Processing and EMC Corporation (Hopkinton, MA). Under this agreement, EMC has produced storage subsystems which allow access to data from both SCSI and S/390 channels, and Innovation has produced the software (FDRSOS) which can be used on MVS to backup and restore this data.

EMC provides this capability as an option on their Symmetrix ESP (Enterprise Storage Platform) 3000-series and 5000-series ICDA (Integrated Cached Disk Array) Storage Subsystems. This option does not allow normal MVS access methods to read or write data on the Open System volumes; only special software such as FDRSOS has this ability.

WHY? Why do you want to backup Open System data on your MVS mainframe, when there are numerous alternatives? The reasons are numerous:

- Your MVS operations staff already has experience and procedures for taking and managing backups, both onsite and offsite.
- Your MVS tape management system provides automatic management of backup tapes, including retention, scratch pools, and automatic offsite vaulting.
- You may have automated tape libraries on your MVS mainframe, allowing for "lights-out" off-shift backups.
- Your Open System servers may be located in the MVS computer room. Even if they are not, ESCON channels allow the EMC Storage Subsystem to be accessed by MVS even though it is many kilometers away.
- You can consolidate the backups of data belonging to many Open System CPUs (as well as that of your S/390 data) in one location.
- Backups can be taken at high-speed, not depending on LAN or communication link speeds.
- New technology S/390 tapes, such as IBM Magstar, allow for high-speed backup to a small number of tape volumes.

Compared to most local or LAN-based backup systems, FDRSOS provides:

- Efficiency
- Centralized administration
- Improved resource utilization

200.02 FDRSOS CONCEPTS**EMC
SYMMETRIX**

EMC Symmetrix ESP-capable Storage Subsystems can be formatted as:

- S/390 DASD (3380 and 3390 disk volumes) in CKD (count-key-data) format, attached to S/390 systems via parallel (bus/tag) or ESCON channels.
- SCSI disks in FBA (Fixed Block Architecture) format, accessed by UNIX or PC systems using SCSI channels. Although "Open System" is most commonly used to refer to UNIX systems, EMC disks in FBA format are called Open System disks even when they are used by PC-type systems such as Windows NT.
- A combination of the two in the same Symmetrix subsystem.

Although the FBA format of the SCSI disks is not supported by MVS access methods, EMC does allow Open System disks to be accessed by special software (such as FDRSOS) on the S/390 system as long as appropriate EMC channel adapters are installed and properly configured.

The physical disks within the EMC Symmetrix are divided into "logical disks" according to a user-designed configuration which assigns one or more logical disks of user-chosen size to each physical disk. Each logical disk can be configured as a S/390 disk or a SCSI Open System disk. The association of the logical disks to the actual physical disks in the Symmetrix is transparent to the end user.

Every S/390 logical disk naturally has a S/390 device address assigned so that it can be used by MVS. Every SCSI logical disk has a SCSI address associated with it. However, the SCSI disks can also be assigned a S/390 address as well, allowing them to be accessed by FDRSOS.

Although these are called "logical disks" by EMC, they appear to be physical disks to the S/390 or SCSI interfaces, and IBM documentation for the S/390 and documentation for the Open System hardware/software may refer to them as physical devices. In some Open System documentation, "logical volumes" may have a different meaning altogether.

**DEVICE
ADDRESSING**

The device addresses of S/390 and SCSI logical disks within the Symmetrix are assigned by the EMC representative in consultation with the customer. Naturally, the MVS I/O configuration must be updated to reflect these devices (via SYSGEN, IOCP, or HCD, depending on your level of MVS).

Those Symmetrix addresses which represent S/390-formatted disks (if any are present in the subsystem) can be varied online and used like any normal 3380 or 3390 DASD volume. Backup and management of these disks can be accomplished using Innovation's FDR DASD Management products, including FDR, FASTCPK, FDRREORG, and FDR/ABR, or similar software.

However, those addresses corresponding to the SCSI Open System disks cannot be varied online. They look to the S/390 hardware like FBA disks, which are not supported by MVS, so any attempt to VARY or MOUNT those devices will fail. These are the disks which can be backed up with FDRSOS.

CONTINUED . . .

200.02 CONTINUED . . .

FDRSOS
ACCESS TO
OPEN SYSTEM
DATA

FDRSOS includes special I/O routines to identify EMC Open System devices and to access them even though they are offline to MVS. FDRSOS will identify those devices which might be EMC Symmetrix SCSI disks (offline DASD devices) and will perform several validations to ensure that they truly are this type of device (e.g., that they respond to FBA commands). Once identified, FDRSOS can perform high-efficiency FBA I/O to read and write these disks as required.

FDRSOS
FUNCTIONS

FDRSOS DUMP allows you to:

- backup one or more Open System volumes serially (one at a time). Each backup must be directed to a separate output data set on tape or disk, but it is possible to stack multiple backups on a single tape volume (or multi-volume tape set) as multiple files.
- backup one or more Open System volumes in parallel, outputting to separate data sets. If tape output is used, separate tape drives must be used in order for the backups to execute in parallel.
- backup entire volumes (the usual method) or include only selected data sectors (by absolute sector number ranges).
- create a duplicate backup while reading the disk data only once.

Note that FDRSOS has no knowledge of the allocation of data on the Open System disks and cannot selectively backup only allocated data areas.

FDRSOS RESTORE will restore from the backups created by FDRSOS, restoring entire Open System volumes or optionally selected sectors (by absolute sector number ranges).

FDRSOS ERASE can quickly rewrite all (or selected portions of) an Open System volume with zeros.

FDRSOS can PRINT selected data sectors from Open System volumes, and can scan for user-specified ASCII or hexadecimal data strings and print the sectors containing those strings.

Finally, FDRSOS has a function (LABEL) which allows you to assign volume serials to each of the Open System volumes and two others (VARYON and VARYOFF) which allow you to make the Open System volumes available or unavailable for further FDRSOS processing.

VOLUME
IDENTIFICATION

Open System volumes do not normally have MVS-style 6-character volume serials, but MVS disk operations are oriented around such a volser, so FDRSOS requires that every Open System volume be assigned a volume serial. You may assign any desired valid volume serial to each volume, using the LABEL function of FDRSOS (See Section 210.05). This volume serial will be stored in an area of each Open System volume that EMC has reserved for FDRSOS use (outside of the normal data area of the volume).

Once assigned, this volume serial may be used to identify Open System volumes to FDRSOS in JCL and in control statements.

CONTINUED . . .

200.02 CONTINUED . . .

COORDINATION WITH OPEN SYSTEMS

Proper operation of FDRSOS backup and restore may require coordination with the Open System which is using the volume. For example, it may be necessary to quiesce updates to files on the Open System while the FDRSOS backup is taking place. Restores may require that the volume be dismounted on the Open System and remounted afterwards. FDRSOS includes utility programs which can be executed on the Open Systems to assist with this coordination for use in situations where the Open System does not provide native commands to do the necessary coordination.

Innovation Data Processing's LAN-to-MVS backup product, FDR/UPSTREAM, can initiate programs and batch files on many Open Systems as described in [Section 220.13](#)

COORDINATION WITH OTHER BACKUPS

Depending on the type of data being backed up, you may need to coordinate FDRSOS backups with other backups of the same data to provide all of the recovery that you need. Remember that FDRSOS backups are full-volume backups which are used primarily for full-volume recovery. [Section 220.11](#) goes into detail on the coordination that may be required.

Innovation Data Processing's LAN-to-MVS backup product, FDR/UPSTREAM, can provide the additional backups for most types of data. FDR/UPSTREAM is being enhanced to initiate and record FDRSOS backups so that they can be part of FDR/UPSTREAM backups. [Section 220.13](#) has more details.

SECURITY CONSIDERATIONS

Security for FDRSOS operations is provided by RACF or equivalent third-party security products. Security checking is not the default; security must be enabled in FDRSOS using the FDRZAPOP utility described in [Section 230.10](#). For every Open System volume to be dumped, restored or erased, FDRSOS will perform a security check against the DASDVOL security class, using the volume serial of the EMC Open System volume. FDRSOS will check for READ authorization for backups, and ALTER authorization for restores and erases.

UTILITIES

FDRSOSTC is a backup copy utility, described in [Section 220.01](#). FDRSOS backups are usually written in a format which cannot be copied by normal copy utilities (such as IEBGENER); **however, such utilities may copy FDRSOS backups without any error message but will leave the copy in an unusable format.** It is important that FDRSOSTC be used whenever it is necessary to create a copy of a FDRSOS backup.

Several utility programs designed to run on various Open Systems in support of FDRSOS are provided. They are described in the Appendixes of this manual.

200.03 FDRSOS IMPLEMENTATION

These are the steps necessary to implement FDRSOS backup and restore:

**FDRSOS
INSTALLA-
TION**

Of course, FDRSOS must be installed on your MVS system as described in [Section 230.01](#).

**EMC
SYMMETRIX
INSTALLA-
TION AND
CUSTOMIZA-
TION**

If not already present, a EMC Symmetrix 3000-series or 5000-series storage subsystem must be installed with the proper hardware and microcode necessary to support Open System (SCSI-attached) storage as well as concurrent S/390 access to the Open System volumes. The subsystem might also contain S/390-formatted (CKD) storage volumes; these volumes will not be processed by FDRSOS. Your EMC representative will ensure that the proper hardware and microcode is installed.

Your EMC representative will assist in configuring the SYMMETRIX storage subsystem. You will, of course, decide on the size and number of Open System (SCSI) and S/390 (CKD) volumes to be defined in the subsystem. Every S/390 volume will be assigned a S/390 address, and every Open System volume will be assigned both a SCSI address and a S/390 address.

**UPDATE
MVS I/O
CONFIGURA-
TION**

You must now update your MVS I/O configuration to include the new addresses. Depending on the type of CPU and the level of your MVS system, this might be done with:

- Sysgen
- IOCP (I/O Configuration Program) and MVSCP (MVS Configuration Program)
- HCD (Hardware Configuration Definition) dialog

The EMC SYMMETRIX subsystem will be defined as one or more 3990 control units. Depending on your EMC hardware configuration, the same control unit may be used to access both Open System and S/390 volumes, and the same channel paths (CHPIDs) may be used for both.

The Open System volumes must be defined as normal 3380 or 3390 disks. However, since they are NOT normal disks, any attempt to vary them online will result in error messages. **So, it is very important that the I/O configuration mark them as OFFLINE, so that MVS will not attempt to access them at IPL time.** Remember that FDRSOS will be able access them even though they are offline. They also be marked as SHARED.

CONTINUED . . .

200.03 CONTINUED . . .

For example, in HCD, the MVS options for an Open System device will look similar to:

View Device Parameter / Feature Definition				Row 1 of 6
Command ==>			Scroll ==> PAGE	
Configuration ID		:	CURPIO99	Common Configuration data
Device number		:	01E0	Device type : 3390
Generic / VM device type		:	3390	
Parameter/ Feature	Value	Req.	Description	
OFFLINE	Yes		Device considered online or offline at IPL	
DYNAMIC	Yes		Device supports dynamic configuration	
LOCANY	Yes		UCB can reside in 31 bit storage	
ALTCTRL	No		Separate physical control unit path	
SHARED	Yes		Device shared with other systems	
SHAREDUP	No		Shared when system physically partitioned	
***** Bottom of data *****				

In IOCP/MVSCP, specify OFFLINE=YES and FEATURE=SHARED on the IODEVICE macros for the Open System devices. The following is an example of a Symmetrix attached via 2 ESCON channels (CHPIDs 40 and 41), with 128 CKD (S/390) disk addresses (300-37F) and 128 FBA (Open System) addresses (380-3FF):

```

CNTLUNIT CUNUMBR=008,PATH=(40),LINK=(C0),
          UNIT=3990,UNITADD=((00,256))
CNTLUNIT CUNUMBR=009,PATH=(41),LINK=(C4),
          UNIT=3990,UNITADD=((00,256))
IODEVICE ADDRESS=(300,128),CUNUMBR=(008,009), ** CKD DEVICES
          UNIT=3390,UNITADD=00
IODEVICE ADDRESS=(380,128),CUNUMBR=(008,009), ** FBA DEVICES
          UNIT=3390,UNITADD=80,OFFLINE=YES,FEATURE=SHARED

```

Note that some documentation may indicate that UNITADD=00 is required for ESCON; this is true, but what it really means is that the value for the first IODEVICE definition for a given control unit must be 00. So when varying characteristics must be specified for some devices on a control unit (such as OFFLINE=YES), you can have multiple IODEVICE statements with varying values for UNITADD.

As is normal in a I/O configuration for a XA, ESA, or OS/390 system, the device numbers assigned to the Open System devices are arbitrary numbers which do not have to have any relation to the actual S/390 device addresses. The device numbers assigned will be the unit addresses referred to by the UNIT= operand of the FDRSOS MOUNT statement ([Section 210.08](#)).

You may wish to generate an esoteric unit name which includes the EMC Open System devices, which can be used in DISKx DD statements in FDRSOS JCL. For example, you might assign generic name OPENSYS to the Open System devices, allowing you to code UNIT=OPENSYS in JCL. Likewise, you will probably not want to include the Open System devices in other DASD esoteric names such as SYSDA since they will never be used for normal MVS disk allocations.

CONTINUED . . .

200.03 CONTINUED . . .

**VERIFY
CONNECTIVI-
TY**

Once the updated I/O configuration has been activated (by an IPL or dynamic I/O reconfiguration) and the Symmetrix subsystem has been attached to the proper channel paths, you can perform a simple test to verify that the Symmetrix disk addresses are properly connected (addressable by FDRSOS).

For a normal CKD disk, this could be done by initializing them with ICKDSF or VARYing them online, but neither of these will work for Open System volumes.

One way to verify the connectivity is to use the MVS console command DEVSERV (abbreviated DS). For example, if your Open System volumes were at addresses 1E0-1EF, you could issue the command:

DS PATHS,1E0,16

which would produce a display similar to:

```

      IEE459I 16.49.32 DEVSERV PATHS 321
UNIT DTYPE          M CNT VOLSER  CHPID=PATH STATUS
01E0,3390           ,F,000,E#01E0,20=D 21=<
01E1,3390           ,F,000,E#01E1,20=+ 21=<
01E2,3390. . .      ,F,000,E#01E2,20=& 21=<
      . . .
***** SYMBOL DEFINITIONS
F = OFFLINE
< = PHYSICALLY UNAVAILABLE
+ = PATH AVAILABLE
& = RESERVED TO ANOTHER PATH/GROUP
* = LOGICALLY OFF, PHYSICALLY ON
- = LOGICALLY OFF, PHYSICALLY OFF

```

The status of each path (CHPID) to each device is shown. A good connection is indicated by:

- + path available
- & path available but the device is currently reserved to another system, such as a AIX/6000 system

Problems might be indicated by:

- < path online but device could not be contacted, control unit not responding
- * path available but marked offline for this device by a VARY PATH command
- path physically offline to the hardware. You may be able to vary it online with the command: CF CHP(xx),ONLINE.

Other status codes are possible and usually indicate a problem.

CONTINUED . . .

200.03 CONTINUED . . .

**VERIFY
CONNECTIVI-
TY
(continued)**

So in this example, CHPID 20 is successfully connected to the Symmetrix Open System addresses while CHPID 21 is not. Device 01E2 is connected but reserved to another system.

If one or more CHPIDs have a status of * (logically offline but physically online), it may be the result of a MVS problem or a VARY PATH console command. The VARYPATH command of FDRSOS ([See Section 210.07](#)) may be able to resolve the problem.

When connection problems are indicated, they may be due to:

- errors in the Symmetrix internal configuration
- problems with the Symmetrix channel adaptor
- problems with the physical channel connection
- errors in the MVS I/O configuration

Your EMC representative will assist you in diagnosing these problems but you may also contact Innovation Technical Support for assistance. But before you call, please review [Section 220.15](#) (FDRSOS DIAGNOSTIC TECHNIQUES).

CONTINUED. . .

200.03 CONTINUED . . .

**IDENTIFY THE
OPEN SYSTEM
VOLUMES**

In order to manage the backups created by FDRSOS, you obviously must know what is being backed up. In other words, there must be some way to identify which backups contain which Open System volumes. Unfortunately, this is not straightforward.

On the Open System side (the UNIX system, or Windows NT system, or whatever), the disk volumes are known by their SCSI addresses. However, after initial installation, the SCSI addresses are rarely used and are often not even known to the system users. Disk volumes are assigned some other identification which varies by system type. They may be known by a drive letter (e.g., the "C" drive), by a system-assigned identification (e.g., "hdisk3") or some other identification. Unfortunately, this identification is known only to that system; it is usually not recorded on the disk volume itself so FDRSOS cannot determine what it is.

As described earlier, every Open System volume in the EMC Symmetrix subsystem will have both a SCSI address and a S/390 address. It is your responsibility to maintain documentation showing the correspondence between each S/390 Open System address and the data that has been placed on that volume from the Open System side. Your EMC representative can provide you with a configuration report showing the defined volumes and their addresses.

You may wish to maintain a spreadsheet or table similar to this:

SCSI Addr	S/390 Addr	EMC Drive	EMC Dev#	Size (MB)	FDRSOS volser	Connected System	Open Sys Disk ID	Format	Usage
00	01E0	DA1C0	00	2000	AIX1H1	AIX Sys 1	hdisk1	JFS	Gen Files
01	01E1	DA1C0	01	2000	AIX1H2	AIX Sys 1	hdisk2	Oracle	Database
10	01E2	DA1D0	02	4000	NTADRD	NT A	D:	HPFS	Payroll

STATUS PRINT

FDRSOS contains a function which can assist you with verifying that you have correctly identified the Open System volume mounted on each S/390 device address. The PRINT=STATUS displays will display a great deal of information about the format of each Open System volume processed and the Open System which created it. [Section 200.04](#) contains more detail and examples on PRINT=STATUS.

200.03 CONTINUED . . .

**OPEN
SYSTEM
VOLUME
SERIALS**

FDRSOS requires that every Open System volume be assigned a volume serial number, in standard MVS format (up to 6 alphabetic, numeric or national characters, e.g., OPEN#2 or EMC123). This allows you to create jobstreams which are independent of the actual device addresses, since you specify the volume serials instead of device numbers to the FDRSOS backup and restore jobs.

We recommend that you assign volume serials that are in some way related to the name by which the volume is known on the Open System which uses it. You might use a form of the Open System volume name, or include the SCSI address (Target/Lun) or use the Open System device name. For example, for an AIX hdisk1 on a RS/6000 system, you might call it AIXHD1. For a Sun/Solaris hard disk "/dev/dsk/cat3d0s0" at SCSI Target/Lun 3/01, you might call it C2T3D1.

The Open System volume serials are assigned using the LABEL function of FDRSOS described in [Section 210.06](#). This assignment needs to be done only once for a given volume; the volser is recorded in an area of the EMC volume that is reserved for FDRSOS use. If new Open System volumes are defined, you will need to run FDRSOS LABEL to assign volsers to the new volumes only. Otherwise you only need to run LABEL if:

- you need to change a volume serial
- the EMC hardware has been reconfigured changing the size or location of the Open System volumes so that the original volume serials have been lost
- a new EMC Symmetrix subsystem is replacing the original system (such as a replacement subsystem at a disaster site).

The assigned volume serial is also stored in the MVS UCB (Unit Control Block) of the Open System volume, although the device will continue to appear offline to MVS. This allows the volume to be referenced in JCL and also shows up in many device displays. For example, a MVS "D U" command will show results similar to:

IEE457I 11.02.51 UNIT STATUS				
UNIT	TYPE	STATUS	VOLSER	VOLSTATE
01E0	3390	OFFLINE	EMC123	/RSDNT
01E1	3390	OFFLINE	OPEN04	/RSDNT

However, after an IPL, the volume serials stored in the MVS UCBs are lost. So, after an IPL but before any other FDRSOS functions are run, you must execute the VARYON function of FDRSOS (see [Section 210.07](#)) to read the volume serials from each Open System volume and reinsert them in the MVS UCBs. Innovation recommends that you run the VARYON step as an automatically executed job or started task at IPL time or as the first step of any FDRSOS job.

**FDRSOS
JOBSTREAMS**

Finally, you must create the batch jobstreams to execute FDRSOS to backup and restore your Open System volumes, as described in the remainder of this manual. The examples in [Sections 210.20-23](#) are a good starting point.

200.04 FDRSOS VOLUME STATUS DISPLAYS

Many FDRSOS functions accept an operand of PRINT=STATUS (see the statement descriptions in [Section 210](#)). If specified, FDRSOS will examine each Open System volume being processed, and attempt to identify the format of the volume using internal indicators and operating system footprints on the volume. If the volume type is successfully identified, FDRSOS will report the format and the type of Open System which created the volume (e.g., IBM AIX, HP-UX, SUN SOLARIS, NOVELL, OS/2, Windows NT, etc.). If applicable, it will display volume or volume group identifications, volume group information, UNIX mount points, and other information; the details will vary depending on the type of Open System.

If the additional operand DIR is specified, e.g., PRINT=(STATUS,DIR), for some disk formats it will also display file information from the root directory on the volume.

NOTE: The volume information displayed by FDRSOS is based on Innovation's current understanding of the format of volumes created by each of the Open Systems supported, as obtained by research and experimentation. Innovation will update the PRINT=STATUS display in subsequent releases of FDRSOS based on experience. If FDRSOS does not recognize the format of your Open System volumes ("UNKNOWN VOLUME TYPE") or displays incorrect information, please contact Innovation so that we can investigate and correct the problem.

Following are examples of PRINT=(STATUS,DIR) displays for some of the Open System formats currently supported by FDRSOS. Some additional formats (e.g., AT&T/NCR) are supported with similar displays. The first FDR216 message always displays the MVS unit address, FDRSOS pseudo volume serial, and Symmetrix device number (subsystem ID and internal logical disk number).

SUN SOLARIS

```
FDR216  STATUS OF SOS SCSI VOLUME UNIT=01EC VOL=E#01EC - SYM-NUMBER=X'00F0-0026'
FDR216  - UNIT=01EC IS AN SUN SOLARIS VOLUME WITH A VOLUME NAME--->NO NAME
FDR216  - UNIT=01EC -WITH A LABEL--->EMC-SYMMETRIX-5063 cyl 4428 alt 2 hd 15 sec 64
FDR216  - UNIT=01EC -SLICE 00 (ROOT) 128 MB SIZE MOUNT POINT--->/
FDR216  - UNIT=01EC -SLICE 01 (SWAP) 128 MB SIZE
FDR216  - UNIT=01EC -SLICE 02 (BACKUP) 2,076 MB SIZE--->/
FDR216  - UNIT=01EC -SLICE 06 (USR) 671 MB SIZE--->/use
FDR216  - UNIT=01EC -SLICE 06 (USR) 257 MB SIZE--->/export/home
```

For SUN Solaris systems, the volume is defined in "slices". Note that the "backup slice" defines the entire volume, so in this example the EMC Symmetrix volume is defined as 2GB. The "usr" slice is the area usable for data files.

The label displayed is a label written by Solaris when the volume was formatted. The text is obtained from the Symmetrix subsystem at that time and shows the Symmetrix microcode level and device size. Note that the microcode level may not be the current level if it has been updated since the volume initialization.

AT&T/NCR
(SYSTEM V)

```
FDR216  STATUS OF SOS SCSI VOLUME UNIT=01E6 VOL=E#01E6 - SYM-NUMBER=X'00F0-0027'
FDR216  - UNIT=01E6 - SYSTEM V UNIX PARTITION WITH A VOLUME NAME-volumec
FDR216  - UNIT=01E6 -SLICE 00 (ROOT) 123 MB SIZE MOUNT POINT--->/
FDR216  - UNIT=01E6 -SLICE 01 (SWAP) 128 MB SIZE
FDR216  - UNIT=01E6 -SLICE 02 (BACKUP) 2,076 MB SIZE MOUNT POINT--->/
FDR216  - UNIT=01E6 -SLICE 06 (USR) 671 MB SIZE MOUNT POINT--->/usr
FDR216  - UNIT=01E6 -SLICE 07 (ALTTRK) 257 MB SIZE MOUNT POINT--->/export/home
FDR216  - UNIT=01EC -SLICE 06 (USR) 257 MB SIZE--->/export/home
```

AT&T and NCR systems run SYSTEM V UNIX, which has a disk structure similar to SUN Solaris.

200.04 CONTINUED. . .

■ IBM AIX/6000

```

FDR216  STATUS OF SOS SCSI VOLUME UNIT=01FA VOL=E#01FA - SYM-NUMBER=X'00F0-003A'
FDR216  - UNIT=01FA IS AN IBM AIX VOLUME WITH A PVID---->00024141-0340F21E
FDR216  - UNIT=01FA IS VOLUME 1 OF VOL GROUP ID VGID--->00024141-0340F764
FDR216  - UNIT=01FA GROUP HAS 4 LOGICAL VOLUMES ON 2 PHYSICAL DISKS
FDR216  - UNIT=01FA 259 PARTITIONS USED ON VOLUME OF 8 MBYTES EACH
FDR216  - UNIT=01E4 MIRROR COPY OF VOLUME->testmirr
FDR216  - UNIT=01E4 -WITH A MOUNT POINT-->/testmirr
FDR216  - UNIT=01FA LOGICAL VOLUME NAME-->testfs01
FDR216  - UNIT=01FA -WITH A MOUNT POINT-->/testfs01
FDR216  - UNIT=01FA LOGICAL VOLUME NAME-->testmirr
FDR216  - UNIT=01FA -WITH A MOUNT POINT-->/testmirr

```

On AIX/6000 UNIX systems (for the IBM RS/6000), physical volumes (Symmetrix logical volumes) may be defined as one or more AIX logical volumes on a volume group. The volume group can consist of one or more physical volumes and the logical volumes may be spread across those physical volumes. Logical volumes may also be mirrored, where the data is recorded multiple times on the physical volumes for recoverability; however, the hardware mirroring facility of the Symmetrix is a superior way of accomplishing the same thing. The physical volumes are divided into partitions of fixed size; one or more partitions are assigned to logical volumes or mirrors as needed.

In this example, the PVID (physical volume ID) is assigned by AIX when the volume is initialized; it is a combination of the disk serial number and a timestamp. All volumes in the AIX volume group are assigned a similar VGID (volume group ID). The number of physical volumes (2) and logical volumes (4) in the volume group are displayed, as well as the partition size (8MB) and the number of partitions on this physical volume. The name and mount point of each logical volume on this physical volume are shown; logical volumes which are AIX mirrors are identified.

■ HP/UX

```

FDR216  STATUS OF SOS SCSI VOLUME UNIT=01E0 VOL=E#01E0 - SYM-NUMBER=X'00F0-0020'
FDR216  - UNIT=01E0 IS AN HP-UX VOLUME WITH A PVID----->78341E7E-330CBB81
FDR216  - UNIT=01E0 IS VOLUME 1 OF VOL GROUP ID VGID-->78341E7E-330CBB83
FDR216  - UNIT=01E0 GROUP HAS 1 LOGICAL VOLUMES ON 1 PHYSICAL DISKS
FDR216  - UNIT=01E0 - WITH A VERITAS FILESYSTEM

```

HP/UX UNIX systems have volume groups and logical volumes similar to IBM AIX/6000 systems (see description above). FDRSOS displays the PVID (physical volume ID) and VGID (volume group ID) assigned by HP/UX. Mount points and root directories may also be displayed.

However, unlike AIX, HP/UX has no convenient way of displaying the PVID and VGID of disks. Innovation has provided a utility which can be run on HP/UX to display the PVID and VGID of every online disk, allowing you to easily match the data displayed by PRINT=STATUS with the disk as seen by HP/UX. This utility is on the diskette supplied with FDRSOS and is called HPPVDISP.TAR; simply uncompress the program and execute HPPVDISP under HP/UX.

Veritas filesystems have no mount points to display. If the HP/UX volume has a standard HP file system, mount points are displayed.

200.04 CONTINUED. . .

**NOVELL
NETWARE**

```

FDR216  STATUS OF SOS SCSI VOLUME UNIT=01EF VOL=E#01EF - SYM-NUMBER=X'00F0-002F'
FDR216  - UNIT=01EF HAS A PC/UNIX PARTITION TABLE WITH A SIGNATURE OF 147E84ED
FDR216  - UNIT=01EF PARTITION 1 OCCUPIES SECTOR 63 TO 2,120,579 - 1,035 MB SIZE
FDR216  - UNIT=01EF --OEM ID-->NOVELL TYPE-->NETWARE
FDR216  - UNIT=01EF --NETWARE VOLUME NAME-->DATABASE 1 - 100 MB SIZE
FDR216  - UNIT=01EF --NETWARE VOLUME NAME-->DATABASE 2 - 10 MB SIZE
FDR216  - UNIT=01EF --NETWARE VOLUME NAME-->SPARE 1 - 4 MB SIZE
FDR216  - UNIT=01EF --NETWARE VOLUME NAME-->SPARE 2 - 5 MB SIZE
FDR216  - UNIT=01EF --NETWARE VOLUME NAME-->DATABASE 3 - 25 MB SIZE

```

Novell Netware volumes may be partitioned, so FDRSOS displays the number and size of each partition (this example has only one partition). Within each partition it displays the partition format (Novell Netware) and the Netware volumes in that partition with their sizes.

WINDOWS NT

```

FDR216  STATUS OF SOS SCSI VOLUME UNIT=01FD VOL=E#01FD - SYM-NUMBER=X'00F0-003D'
FDR216  - UNIT=01FD HAS A PC/UNIX PARTITION TABLE WITH A SIGNATURE OF 147E84E7
FDR216  - UNIT=01FD PARTITION 1 OCCUPIES SECTOR 32 TO 2,119,679 - 1,035 MB SIZE
FDR216  - UNIT=01FD --OEM ID-->NTFS TYPE-->NTFS VOLID-->D4FD-D067 NAME-->DATABASE1

```

Windows NT volumes may be partitioned, so FDRSOS displays the number and size of each partition (this example has only one partition). Within each partition it displays the partition format (NTFS).

200.04 CONTINUED. . .

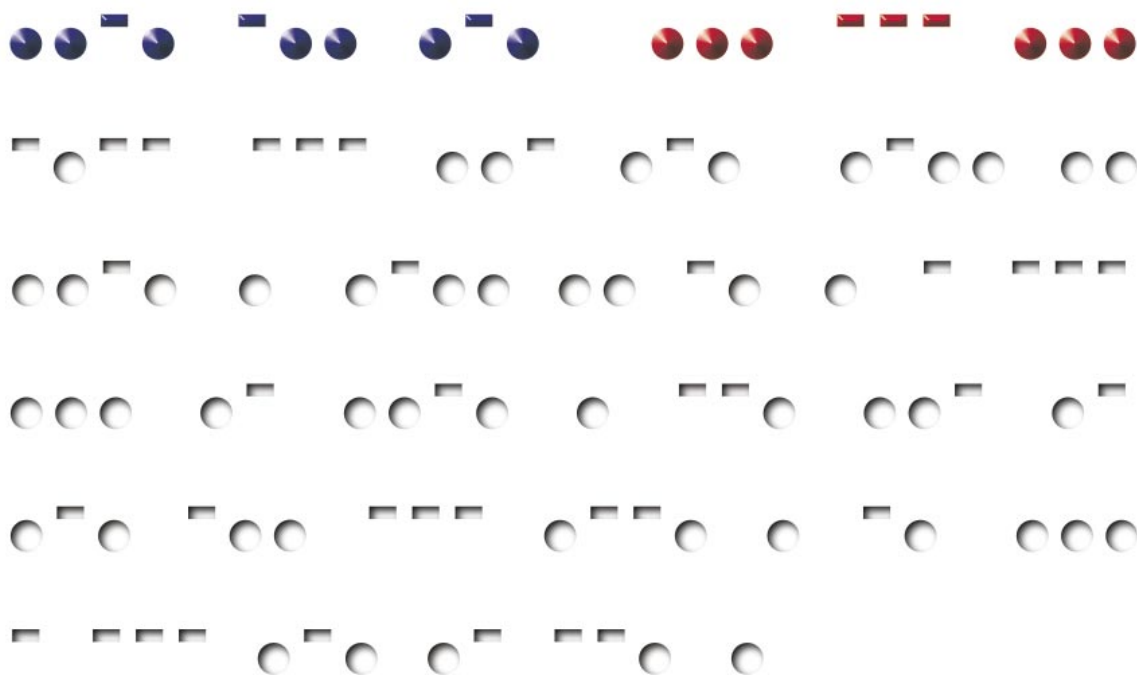
OS/2

```

FDR216 STATUS OF SOS SCSI VOLUME UNIT=01EA VOL=E#01EA - SYM-NUMBER=X'00F0-002A'
FDR216 - UNIT=01EA HAS A PC/UNIX PARTITION TABLE WITH A SIGNATURE OF 147E84E9
FDR216 - UNIT=01EA ->LINKAGE PARTITION ON SECTOR 2,048 TO 2,119,679 - 1,034 MB SIZE
FDR216 - UNIT=01EA PARTITION 1 OCCUPIES SECTOR 2,080 TO 206,847 - 100 MB SIZE
FDR216 - UNIT=01EA --OEM ID-->OS2 20.0 TYPE-->HPFS VOLID-->A698-0C14 NAME-->PAYROLL 1
FDR216 - UNIT=01EA --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216 - UNIT=01EA 01/14/97 15.59.15 8 20 QUERY.EXE
FDR216 - UNIT=01EA 01/14/97 15.59.15 8 20 README.TXT
FDR216 - UNIT=01EA 01/14/97 16.47.49 <DIR> 50 payroll
FDR216 - UNIT=01EA -END OF DIRECTORY ENTRIES
FDR216 - UNIT=01EA ->LINKAGE PARTITION ON SECTOR 206,848 TO 227,327 - 10 MB SIZE
FDR216 - UNIT=01EA PARTITION 2 OCCUPIES SECTOR 206,880 TO 227,327 - 10 MB SIZE
FDR216 - UNIT=01EA --OEM ID-->OS2 20.0 TYPE-->HPFS VOLID-->26E5-0815 NAME-->PAYROLL 2
FDR216 - UNIT=01EA --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216 - UNIT=01EA -END OF DIRECTORY ENTRIES
FDR216 - UNIT=01EA ->LINKAGE PARTITION ON SECTOR 227,328 TO 1,251,327 - 500 MB SIZE
FDR216 - UNIT=01EA PARTITION 3 OCCUPIES SECTOR 227,360 TO 1,251,327 - 500 MB SIZE
FDR216 - UNIT=01EA --OEM ID-->OS2 20.0 TYPE-->HPFS VOLID-->26F1-4815 NAME-->HUMAN RES
FDR216 - UNIT=01EA --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216 - UNIT=01EA -END OF DIRECTORY ENTRIES
FDR216 - UNIT=01EA ->LINKAGE PARTITION ON SECTOR 1,251,328 TO 1,353,727 - 50 MB SIZE
FDR216 - UNIT=01EA PARTITION 4 OCCUPIES SECTOR 1,251,360 TO 1,353,727 - 50 MB SIZE
FDR216 - UNIT=01EA --OEM ID-->IBM 20.0 TYPE-->FAT VOLID-->E272-C815 NAME-->NO NAME
FDR216 - UNIT=01EA --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216 - UNIT=01EA 01/16/97 14.34.34 <VOLUME LABEL> 08 FAT-VOL03
FDR216 - UNIT=01EA 01/16/97 14.47.36 10 20 TESTERGG.GGG

```

OS/2 volumes may be partitioned, so FDRSOS displays the number and size of each partition and the volume ID on that partition (the volume ID consists of a "volume serial" assigned by OS2 and the optional "volume label" specified by the user when the partition was formatted. Within each partition it displays the partition format (OS2 HPFS or IBM FAT). If the DIR option was specified ("PRINT=(STATUS,DIR)") the contents of the root directory in each partition is displayed (files and subdirectory names). In this example, partitions 2 and 3 have been formatted but contain no files



Installation and User's Guide

THIS PAGE WAS INTENTIONALLY LEFT BLANK

210.01 FDRSOS JOB CONTROL REQUIREMENTS

To execute FDRSOS, the following JCL statements are required. Please review the examples ([sections 210.20-23](#)) for a better understanding of the JCL requirements of FDRSOS.

**STEPLIB or
JOB LIB DD
STATEMENT**

If required, specifies the job library in which FDRSOS resides. The library must be authorized.

**EXEC
STATEMENT**

Specifies the program name (PGM=FDRSOS), region requirement (REGION=nnnK or nM), and optional PARM= field which may contain the first FDRSOS control statement. For example,

```
//SOS EXEC PGM=FDRSOS,REGION=0M,PARM='DUMP TYPE=FULL,RESERVE=YES'
```

The parm may also contain the second control statement, separated from the first by a slash (/), e.g.,

```
PARM='VARYON TYPE=SOS/MOUNT UNIT=03*'
```

Note that there must be no space immediately before the slash.

Additional control statements, if any, must be contained in the SYSIN data set.

FDRSOS acquires most of its storage below the 16M line, in the area specified by the REGION= operand.

For each concurrent DUMP and PRINT operation, FDRSOS requires about 200K plus the storage required for the number of 36K buffers specified by the BUFNO= operand. With the recommended default of BUFNO=MAX, 30 buffers are acquired (30*36=1080K). If COMPRESS= is used, an additional 100K is needed for each concurrent backup.

So, FDRSOS requires about 1300K for each concurrent backup (1400K with COMPRESS). If multiple concurrent backups are run in the same step because MAXTASKS=n is specified, multiply the region requirement by "n". You must specify a REGION= value at least as large as this requirement; if you do not, some operations may fail. There is no penalty for specifying a region larger than required; FDRSOS will still use only what it needs. The largest value you can specify for REGION= is different at every installation but is usually 6M or more. Innovation recommends that you specify REGION=0M so that FDRSOS has the maximum below-the-line region available.

For RESTORE operations, FDRSOS requires about 1400K for each concurrent restore. If MAXTASKS=n is specified, the region must be 1400K*n.

ERASE, LABEL, VARYON, and VARYOFF operations require about 500K.

**SYSPRINT DD
STATEMENT**

Specifies the output message data set; it is required. It is usually a SYSOUT data set but if it is assigned to a data set on tape or disk, this DD must specify DISP=MOD.

CONTINUED . . .

210.01 CONTINUED . . .

**SYSPRINx DD
STATEMENT**

Specifies the output data set for dump messages related to the matching TAPEx DD when the MAXTASKS= option is used; there must be a SYSPRINx for every TAPEx. It is usually a SYSOUT data set but if it is assigned to a data set on tape or disk, this DD must specify DISP=MOD. If MAXTASKS= is not used, the SYSPRINx DD statements can be omitted since all messages will be written to SYSPRINT.

**FDRSUMM
DD
STATEMENT**

(Optional) if present, FDRSOS will write one-line messages for each Open System volume dumped, restore, or printed, giving result codes, elapsed time, and byte counts. Usually a SYSOUT data set. Innovation recommends including FDRSUMM.

**SYSUDUMP
DD
STATEMENT**

Specifies the abend data set. Usually a SYSOUT data set. A SYSUDUMP DD statement should always be included to assist in error diagnosis.

If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced: //ABNLIGNR DD DUMMY

CONTINUED . . .

210.01 CONTINUED . . .

**TAPE_x DD
STATEMENT**

Used to specify the output data set for DUMP, the input data set for RESTORE and the print data set for PRINT. "x" may be any alphanumeric character (0-9, A-Z). Multiple TAPE_x DD statements may be present in the FDRSOS step JCL; a unique value for 'x' must be used for each of them (e.g., TAPE1, TAPE2, etc.)

For DUMP Operations:

Specifies a tape or sequential disk data set to which the backup will be written. You must provide a TAPE_x DD statement for each Open System volume to be backed up in this step. If DISK_x DD statements (described below) are used, the TAPE_x DD will receive the backup of the volume specified by DISK_x. If MOUNT statements are used, the TAPEDD=_x operand on the MOUNT statement controls which volume will be directed to each backup data set (if only a single volume is backed up with MOUNT, TAPEDD= can be omitted and the backup will be written to the first or only TAPE_x DD in the step).

If MAXTASKS= is specified with a value greater than 1, FDRSOS will attempt to attach that many concurrent backup subtasks, but may postpone some of them if it detects that they require a tape drive in use by another backup (see the notes on UNIT=AFF and VOL=REF below).

DUMMY is supported, for **testing** purposes only.

You must provide all the JCL parameters required to allocate and catalog the backup data set on disk or tape, which may include some or all of: DSN=, UNIT=, VOL=, SPACE=, and DISP=(NEW,CATLG). For tape, a volume count should be specified since the default is only 5 tape volumes, e.g., VOL=(,,255).

DCB parameters are not required and should be omitted. However, IDRC (tape unit hardware compression, available on most tape cartridge drives) can be requested by adding DCB=TRTCH=COMP to your DD statement; IDRC compression may be the default depending on local MVS options. For tapes attached by ESCON channels, Innovation recommends use of IDRC compression instead of FDRSOS software compression (the COMPRESS= option).

```
Examples : //TAPE1DDDSN=PROD.OPEN.BACKUP1,UNIT=TAPE,
           //          VOL=(,,255),DISP=(NEW,CATLG)
           //TAPE2DDDSN=PROD.OPEN.BACKUP2,UNIT=DISK,
           //          SPACE=(CYL,(500,100),RLSE),DISP=(NEW,CATLG)
```

CONTINUED . . .

210.01 CONTINUED . . .

**TAPEx DD
STATEMENT
(continued)**

For tape backups, UNIT=AFF or VOL=REF may be specified, referencing another TAPEx DD statement, to reduce the number of tape drives used in the step. UNIT=AFF=TAPEx will cause MVS to allocate the same tape drive for both DD statements, but will call for separate output tapes when each DD is opened. VOL=REF=*.TAPEx with LABEL=n can be used to stack multiple backup files on the same tape, providing more complete utilization of the tape volumes (which may be important as new technology increases tape volume capacity). FDRSOS will automatically recognize that multiple TAPEx DDs point to the same tape drive and will serialize operations on that drive so that only one backup is directed to that drive at a time.

Examples: `/* The following creates 2 backups on 2 different
/* tape volumes using the same tape drive. This may not
/* fully utilize the tape volumes but will allow for
/* concurrent restores from these backups.
//TAPE1 DD DSN=PROD.OPEN.BACKUP1,UNIT=TAPE,
// VOL=(,,,255),DISP=(NEW,CATLG)
//TAPE2 DD DSN=PROD.OPEN.BACKUP2,UNIT=AFF=TAPE1,
// VOL=(,,,255),DISP=(NEW,CATLG)

/* The following creates a multi-file (and possibly
/* multi-volume) tape containing 3 backups.
//TAPE3 DD DSN=PROD.OPEN.BACKUP3,UNIT=TAPE,
// VOL=(,,,255),DISP=(NEW,CATLG)
//TAPE4 DD DSN=PROD.OPEN.BACKUP4,LABEL=2,
// VOL=REF=*.TAPE3,DISP=(NEW,CATLG)
//TAPE5 DD DSN=PROD.OPEN.BACKUP5,LABEL=3,
// VOL=REF=*.TAPE4,DISP=(NEW,CATLG)`

For RESTORE Operations:

Specifies a backup dataset on tape or disk from which the data is to be restored. The backup must, of course, be a backup created by FDRSOS. You may include multiple TAPEx DD statements in order to restore multiple Open System volumes. A DISKx DD statement or the TAPEDD=x operand on a MOUNT statement is required to define which backup is to be restored to which volume.

Example: `//TAPE1 DD DSN=PROD.OPEN.BACKUP1,DISP=OLD`

However, if the backup data set is cataloged, RESTOREs can also be done without TAPEx DD statements. The BACKUPDSN= operand on the MOUNT statement can be used to specify the name of a cataloged data set from which an Open System volumes is to be restored; FDRSOS will dynamically allocate that data set.

If MAXTASKS= is specified with a value greater than 1, FDRSOS will attempt to attach that many concurrent restore subtasks, but may postpone some of them if it detects that they require a tape drive in use by another restore. You can use UNIT=AFF (see notes under DUMP above) to assign two or more restores to use the same type drive, but MVS may do this automatically if it detects that two input data sets required the same tape volume. MAXTASKS= is ignored if BACKUPDSN= is used to allocate the input data sets instead of TAPEx DD statements; restores with BACKUPDSN= are always done serially within a given job but you can execute multiple restore job to get the same effect.

CONTINUED . . .

210.01 CONTINUED . . .

**TAPEx DD
STATEMENT
(continued)**

For PRINT Operations:

Specifies the destination of the printed output from the PRINT statement. This is usually SYSOUT, but it may be directed to a disk or tape data set.

Example: //TAPE1 DD SYSOUT=*

For ERASE Operations:

Can be omitted if MAXTASKS=n is not specified and MOUNT statements are used to identify the volumes to be erased. However, if MAXTASKS=n is included, a TAPEx DD pointing to DUMMY and a SYSPRINx DD for messages must be included for each disk to be erased. If DISKx DDs are used to identify the volume, matching TAPEx DDs pointing to DUMMY must also be included.

Example: //TAPE1 DD DUMMY

For LABEL, VARYON and VARYOFF Operations:

TAPEx DD statements are not required and should be omitted.

**TAPExx DD
STATEMENT**

For a DUMP operation only, specifies a second backup data set. The TAPExx data set will contain a backup identical to that written to TAPEx. For example, if a backup is being written to TAPE6, the inclusion of a TAPE66 DD statement will cause a second duplicate backup file to be produced. A given TAPExx DD statement will be ignored if there is no TAPEx DD with the same "x" character.

However, these restrictions exist:

- if TAPEx is a backup data set on disk (backup to disk), then TAPExx cannot also point to a backup data set on disk. In other words, you cannot create 2 simultaneous disk backups of an Open System volume. You can backup TAPEx to tape and TAPExx to disk, or vice versa.

210.01 CONTINUED. . .

**DISKx DD
STATEMENT**

Optionally used to specify the Open System volume to be dumped, restored, erased or printed (a FDRSOS MOUNT statement can also be used to identify the volumes to be processed). If used, the format will be:

```
//DISK1 DD UNIT=unitname,VOL=SER=volser,DISP=OLD
```

"unitname" is either a generic name, such as 3390, or an esoteric name assigned during your I/O configuration, such as OPENSYS ([see Section 200.03](#)), and "volser" is the volume serial assigned to the Open System volume using the FDRSOS LABEL function ([see Section 210.06](#)).

Warning: if an IPL of your MVS system has occurred since the FDRSOS LABEL function was run for a given Open System volume, you **must** run the FDRSOS VARYON function for each of the Open System devices before the volume serial can be used in JCL ([see Section 210.07](#)). Innovation recommends that you either run the VARYON function as an automatically executed job or started task at IPL time, or as the first step of any FDRSOS backup/restore job.

The "x" in DISKx must match a corresponding TAPEx DD statement. For DUMP, the Open System volume specified by DISKx will be backed up to TAPEx (and optionally TAPExx). For RESTORE, the DISKx volume will be restored from the TAPEx backup. For ERASE, the DISKx volume will be erased. For PRINT, the DISKx volume is printed on the TAPEx output data set. DISKx DD statements must be omitted for LABEL, VARYON and VARYOFF functions.

If DISKx DD statements are used, the Open System volumes will be processed in the order that the DISKx DDs appear in the FDRSOS JCL. If MOUNT statements are used, the volumes will be processed in the order that their UCBs are found in the MVS system, which may be difficult to predict. If the order of processing is important (such as when you are creating multiple files on a tape volume), DISKx DD statements are the only way to guarantee the order.

Note: Although the EMC microcode and cache will optimize performance in most cases, degradation due to excessive movement of the disk heads may occur when you are concurrently processing logical volumes which reside on the same physical disk within the EMC subsystem. For best performance, you should be aware of the physical location of the logical volumes involved and plan your backups and restores accordingly. You may need to segregate FDRSOS operations into multiple jobs run serially to enforce this.

Note: the MVS system messages for a FDRSOS step will contain an allocation for a DDname of "D#FDRSOS" as a temporary data set on your system residence volume. This is normal; it is for allocation purposes only and no real data set is allocated.

**SYSIN DD
STATEMENT**

Specifies the control statement data set. Usually an input stream or DD * data set. It may be DUMMY if all necessary control statements are included in the PARM= on the EXEC statement.

210.02 FDRSOS DUMP STATEMENT

DUMP D	TYPE=FULL PARTIAL ,BLKSIZE=nnnnn ,BUFNO= <u>MAX</u> nn ,COMPRESS=ALL COPY1 COPY2	,MAXERR=nnnn ,MAXTASKS= <u>1</u> n ,PRINT=(STATUS,DIR) ,PRINT=UCB ,RESERVE=YES <u>NO</u>
-----------	---	--

**DUMP
STATEMENT**

This statement requests a backup operation. It must be the first statement in the input; only one DUMP statement is allowed per execution. DUMP must be followed by one or more MOUNT statements to specify the Open System volumes to be backed up unless DISKx DD statements are used to identify the volumes ([See Section 210.01](#)).

If you have security checking enabled (via FDRZAPOP as shown in [Section 230.10](#)), DUMP requires that the user have at least READ authority to the Open System volume serial under the DASDVOL security class.

OPERANDS TYPE=

Specifies the type of backup and must be specified on the DUMP Statement.

FULL - requests that entire Open System volumes be backed up. This will backup every data sector on the selected volumes, including those which may not be currently allocated to a file.

PARTIAL - requests that selected sections of Open System volumes be backed up. One or more SELECT statements must be present. Partial backups should not be done unless you have a detailed knowledge of the structure and layout of data on the disk volumes.

BLKSIZE=

specifies the size in bytes of the largest block to be written on the backup tape. The value may range from 6000 to 61440. BLKSIZE= is ignored when outputting to a sequential DASD data set; the blocksize will be the maximum size of a half-track record on the output DASD device type. Regardless of the actual blocksize used, the tape label or disk DSCB of the backup data set will show RECFM=U and BLKSIZE=32760 because of MVS restrictions, so this cannot be used to determine the actual blocksize of the backup. If a BLKSIZE= value of 26000 or less is specified, the value of BUFNO= may be automatically decreased.

The default is 61440.

Note: Innovation recommends that you do not override the blocksize, for best performance and maximum tape and DASD utilization. However, tapes with blocks over 32760 in length may not be copied by standard copy utilities and usually cannot be transmitted with data transfer utilities; if this is necessary, BLKSIZE=32760 can be used to create a compatible backup tape with minimal impact on FDRSOS performance.

[See Section 220.01](#) for documentation on FDRSOSTC, a utility to copy FDRSOS tapes with any blocksize.

210.02 CONTINUED. . .

- BUFNO=** specifies how many buffers will be used for dumping each Open System volume. Each buffer is 36K in length. The buffers acquired will be divided into 2 sets in order to overlap input and output I/O operations; each disk I/O will read into one half of the buffers. Reducing the number of buffers will reduce the amount of below-the-line storage required for each concurrent backup, but will also reduce the efficiency of the backup and increase the elapsed time.
MAX - specifies that 30 buffers will be acquired.
nn - the specified number of buffers is acquired. The value may be from 1 to 30 but it will be rounded up to the next higher even number. Values over 30 are treated as 30.
The default is MAX (30). Innovation recommends that you do not override the default.
- COMPRESS=** **ALL** – specifies that FDRSOS is to compress the output data on the backup file for both copies (TAPEX and TAPEXX if present).
COPY1 – specifies that only the backup data on TAPEX DD statements will be compressed.
COPY2 – specifies that only the backup data on TAPEXX DD statements, if present, will be compressed.
Compression of either or both copies will add 100K to the region required for each concurrent backup.
Default is NONE.
COMPRESS is recommended for backups to disk files, and for tape backups to tapes attached on parallel (bus/tag) channels. For tapes attached on ESCON (serial fiber optic) channels, use of IDRC (tape hardware compression) usually results in better performance.
Note: COMPRESS is not supported in the current release. It will be supported in a future release. If you are planning on using this feature, please contact Innovation.
- MAXERR=** Specifies the number of disk errors that are permitted by FDRSOS prior to abending the DUMP operation. MAXERR may specify a value from 1 to 9999 errors; MAXERR=1 forces termination immediately on any error. Each error will be indicated by a message and other diagnostic information; such errors usually indicate that some data was omitted from the backup which may make the backup unusable.
Default is 20 errors.

210.02 CONTINUED. . .

MAXTASKS= Specifies the maximum number of Open System volumes that will be dumped concurrently in this step. The value may be from 1 to 9 but the actual number of concurrent dumps will be no greater than the number of TAPEX DD statements in this step's JCL (if multiple TAPEX DD statements point to the same tape drive, one only concurrent dump will use that drive at any time).

The default is 1. However, if MAXTASKS= is omitted all backup messages will be printed on SYSPRINT while if MAXTASKS=n is specified (even if "n" is 1), messages are directed to the SYSPRINx DD corresponding to the TAPEX DD associated with the backup (see Section 210.01).

PRINT= **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being backed up. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

(STATUS,DIR) - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

UCB - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

Note: PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g., PRINT=(STATUS,DIR),PRINT=UCB.

RESERVE= specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being backed up.
YES – the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the backup is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the backup will fail immediately.
NO – no RESERVE is issued.

The default is NO.

Note: Innovation does not recommend use of RESERVE=YES for backups unless you **must** prevent other MVS and/or SCSI systems from accessing the Open System volumes during the backup.

210.03 FDRSOS RESTORE STATEMENT

RESTORE R	TYPE=FULL PARTIAL ,CONFMESS= <u>YES</u> NO ,CPYVOLID=YES <u>NO</u> ,MAXERR=nnnnn	,MAXTASKS=n ,PRINT=(STATUS,DIR) ,PRINT=UCB ,RESERVE= <u>YES</u> NO ,VOLRESET= <u>YES</u> NO
--------------	---	---

**RESTORE
STATEMENT**

This statement requests a restore operation. It must be the first statement in the input; only one RESTORE statement is allowed per execution. RESTORE must be followed by one or more MOUNT statements to specify the Open System volumes to be restored unless DISKx DD statements are used to identify the volumes ([See Section 210.01](#)).

If you have security checking enabled (via FDRZAPOP as shown in [Section 230.10](#)), RESTORE requires that the user have ALTER authority to the Open System volume serial under the DASDVOL security class.

OPERANDS

TYPE=	Specifies the type of restore and must be specified on the RESTORE Statement. FULL – requests that the entire contents of each backup data set be restored. This may be an entire Open System volume if the entire volume was backed up (DUMP TYPE=FULL) or may be selected portions of a volume if the backup was a partial backup (DUMP TYPE=PARTIAL). PARTIAL – requests that selected sections of the contents of each backup data set be restored. A SELECT statement must be present. Note that data from Open System volumes will always be restored to the same block locations it was backed up from. Partial restores should not be done unless you have a detailed knowledge of the structure and layout of data on the disk volumes.
CONFMESS=	YES – specifies that, before beginning the restore, FDRSOS will request confirmation via a WTOR message (FDRW01) to which the operator must reply. NO – suppresses the WTOR and begins the restore immediately. CONFMESS=NO can be very useful at a disaster recovery site to avoid full volume restores being delayed waiting for an operator response. Default is YES.
CPYVOLID=	YES – FDRSOS will compare the volume serial of the Open System volume which was backed up in the input backup data set to the volume serial of the output disk. If they are different, at the end of the restore FDRSOS will change the volume serial of the output disk to the volume serial from the backup. The volume serial stored in the area of the volume reserved for FDRSOS use will be updated. The volume serial stored in the UCB of the output disk will be updated unless the same volume serial is already set in the UCB of another disk (either online or offline); in that case, the volume serial in the output UCB will be cleared (effectively a VARYOFF of the output volume). NO – the volume serial of the output disk will not be changed, even if it is different from that of the Open System volume on the backup data set. Default is NO.

CONTINUED . . .

210.03 CONTINUED. . .

MAXERR= Specifies the number of disk errors that are permitted by FDRSOS prior to abending the restore operation. MAXERR may specify a value from 1 to 9999 errors; MAXERR=1 forces termination immediately on any error. Each error will be indicated by a message and other diagnostic information; such errors usually indicate that some data was lost from the backup which may make the restored volume unusable.

Default is 20 errors.

MAXTASKS= Specifies the maximum number of Open System volumes that will be restored concurrently in this step. The value may be from 1 to 9 but the actual number of concurrent restores will be no greater than the number of TAPEX DD statements in this step's JCL (if multiple TAPEX DD statements point to the same tape drive, one only concurrent dump will use that drive at any time). MAXTASKS= is ignored if the backup data set is dynamically allocated using the BACKUPDSN= operand on a SELECT statement; it works only if TAPEX DDs are used to identify the backup data sets.

The default is 1. However, if MAXTASKS= is omitted all restore messages will be printed on SYSPRINT while if MAXTASKS=n is specified (even if "n" is 1), messages are directed to the SYSPRINx DD corresponding to the TAPEX DD associated with the restore (see [Section 210.01](#)).

PRINT= **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being restored. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. This information will be displayed both before and after the restore. For more information, see the sample displays in [Section 200.04](#).

(STATUS,DIR) - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

UCB - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

Note: PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,

PRINT=(STATUS,DIR),PRINT=UCB

RESERVE= specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being restored. This is recommended during a RESTORE operation to ensure that the volume will not be accessed until the restore is complete.
YES – the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the restore is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the restore will fail immediately.
NO – no RESERVE is issued.

The default is YES.

Warning: if RESERVE=YES is in effect, you must be sure that no other system (MVS or SCSI) has the Open System volume reserved; otherwise the FDRSOS restore will fail. See [Section 220.11](#) for additional warnings about restore.

CONTINUED. . .

210.03 CONTINUED. . .

VOLRESET= Most Open System volumes contain a volume identifier (usually called a PVID or signature), which is assigned when the volume is formatted. The format and location of the volume identifier varies, depending on the operating system (UNIX, Windows NT, etc.) and can also vary depending on the version of UNIX and the type of file system on the volume. Some volumes may have no volume identifier at all, such as "raw volumes" or other special format volumes. This volume identifier is often used by the Open System in identifying the volume when it is mounted. There may also be other identifiers and volume descriptors, such as logical volume identifiers. Normally, FDRSOS will backup and restore these identifiers, since they are part of the data sectors read and written by FDRSOS. However, when you are restoring an Open System volume to a different location (such as when you are restoring to spare volumes in order to recover individual files, as described in [Section 220.14](#)), you may need to preserve the identifiers currently assigned to the target volumes instead of restoring them from the backup. Please review [Section 220.14](#) for more details.

NO - FDRSOS will attempt to preserve the current identifiers of the target (output) volumes. The target volumes should be preformatted as the same type of volume and file system as the volume being restored from backup.

If FDRSOS is successfully able to identify the type of volume on the target and identify the location of the identifier fields, it will read them from the output disk and substitute them for the identifiers being restored from the backup. This will allow the Open System to see these volumes with unchanged PVID values, although the contents have been restored. FDRSOS may also modify other identifiers (such as logical volume IDs) during the restore. If it cannot identify the IDs of the output disk, it will simply use the IDs from the input volume, with an incremental added to make them unique.

For UNIX and Novell Netware systems, FDRSOS will also attempt to identify the mount point name (UNIX) or logical volume name (Novell) and modify them so that they can be successfully mounted even though the original volumes are still mounted. It will add "_SOS" to mount points and "_S" to logical volume names.

VOLRESET=NO should be used when you are restoring to new volumes when the original volumes are still mounted (such as when preparing for a file-level recovery).

YES - identifier values will be restored from the backups unchanged. VOLRESET=YES should be used when restoring back to the original volumes and when restoring at a disaster site. If you are restoring backups to new volumes (not back on top of the original volumes) and the original volumes are still mounted, this will probably cause the restored volumes to be unusable.

Default is YES.

Warning: do not specify VOLRESET=NO unless you understand the implications as described above and in [Section 220.14](#).

VOLRESET=NO may not support all type of volumes, see [Section 220.14](#) for details.

210.04 FDRSOS PRINT STATEMENT

PRINT	TYPE=FULL PARTIAL	,PRINT=(STATUS,DIR)
P	,MAXERR=nnnnnn	,PRINT=UCB
		,RESERVE=<u>YES</u> NO

**PRINT
STATEMENT**

This statement requests a print operation. It must be the first statement in the input; only one PRINT statement is allowed per execution. PRINT must be followed by one or more MOUNT statements to specify the Open System volumes to be printed unless DISKx DD statements are used to identify the volumes ([See Section 210.01](#)).

The printed output will be directed to the TAPEx DD statement (the first TAPEx DD in the JCL unless DISKx DDs are used or the TAPEDD= operand is coded on the associated MOUNT statement). Each 512 byte data sector will be printed in "dump" format, with 32 bytes of hexadecimal data on each line, and the same data on the right in printable format (translated from ASCII to EBCDIC).

The PRINT function also includes the ability to scan for specified hexadecimal or ASCII strings of data and print only the data sectors containing the specified strings. Certain PRINT options can also be overridden. Details are found in [Section 210.10](#).

CONTINUED . . .

210.04 CONTINUED. . .

OPERANDS	TYPE=	<p>Specifies the type of print and must be specified on the PRINT Statement.</p> <p>FULL – requests that the entire contents of each Open System volume be printed. Since this may be a large amount of printout, this is not recommended unless a SCAN statement (Section 210.10) is present to limit the printout.</p> <p>PARTIAL – requests that selected sections of the contents of each Open System volume be printed. A SELECT statement must be present.</p>
	MAXERR=	<p>Specifies the number of disk errors that are permitted by FDRSOS prior to abending the print operation. MAXERR may specify a value from 1 to 9999 errors. Each error will be indicated by a message and other diagnostic information.</p> <p>Default is 20 errors.</p>
	PRINT=	<p>STATUS - requests that FDRSOS attempt to identify the type of Open System volume being printed. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.</p> <p>(STATUS,DIR) - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.</p> <p>UCB - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.</p> <p>Note: PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,</p> <p>PRINT=(STATUS,DIR),PRINT=UCB</p>
	RESERVE=	<p>specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being printed.</p> <p>YES – the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the restore is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the print will fail immediately.</p> <p>NO – no RESERVE is issued.</p> <p>The default is NO.</p> <p>Note: Innovation does not recommend use of RESERVE=YES unless you must prevent other systems from accessing the Open System volumes during the print.</p>

CONTINUED. . .

210.05 FDRSOS ERASE STATEMENT

```

ERASE                                TYPE=FULL                                ,MAXTASKS=n
                                     ,CONFERASE=YES | NO          ,PRINT=(STATUS,DIR)
                                     ,PRINT=UCB
                                     ,MAXERR=nnnnn              ,RESERVE=YES | NO

```

**ERASE
STATEMENT**

This statement requests an erase operation. It must be the first statement in the input; only one ERASE statement is allowed per execution. ERASE must be followed by one or more MOUNT statements to specify the Open System volumes to be erased unless DISKx DD statements are used to identify the volumes ([See Section 210.01](#)).

ERASE will rewrite all sectors on the target Open System volume as binary zeros.

Obviously this is a very dangerous operation and must be used with care. ERASE might be useful after a disaster recovery test or when replacing a EMC Symmetrix containing Open System data in order to be sure that all company data is removed from the disks.

If you have security checking enabled (via FDRZAPOP as shown in [Section 230.10](#)), ERASE requires that the user have ALTER authority to the Open System volume serial under the DASDVOL security class.

ERASE does not require TAPEX DD statements unless:

- MAXTASKS=n is specified, in which case a TAPEX DD DUMMY and a SYSPRINx DD are required for each volume to be erased (the volume must be identified by either a DISKx DD statement or the TAPEDD=x operand on a MOUNT statement. ERASE messages will be written to SYSPRINx
- A DISKx DD statement is used to identify the volume to be erased, in which case a TAPEX DD DUMMY must be included.

OPERANDS

TYPE= **FULL** - requests that the entire contents of the Open System volume be erased. This is the only value currently supported for ERASE.

CONFERASE= **YES** – specifies that, before beginning the erase, FDRSOS will request confirmation via a WTOR message (FDRW01) to which the operator must reply.
NO – suppresses the WTOR and begins the erase immediately.
 CONFERASE=NO can be very useful at a disaster recovery site when erasing many volumes to avoid waiting for an operator response. However, CONFERASE=NO is not recommended unless speed is of the essence and you are very sure that you have correctly specified the volumes to be erased.

Default is YES.

MAXERR= Specifies the number of disk errors that are permitted by FDRSOS prior to abending the erase operation. MAXERR may specify a value from 1 to 9999 errors; MAXERR=1 forces termination immediately on any error. Each error will be indicated by a message and other diagnostic information.

Default is 20 errors.

210.05 CONTINUED. . .

MAXTASKS= Specifies the maximum number of Open System volumes that will be erased concurrently in this step. The value may be from 1 to 9 but the actual number of concurrent restores will be no greater than the number of TAPEX DD statements in this step's JCL.

The default is 1. However, if MAXTASKS= is omitted all erase messages will be printed on SYSPRINT while if MAXTASKS=n is specified (even if "n" is 1), messages are directed to the SYSPRINx DD corresponding to the TAPEX DD associated with the restore ([see Section 210.01](#)).

PRINT= **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being erased, before the erase begins. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. This information will be displayed both before and after the erase. For more information, see the sample displays in [Section 200.04](#).

(STATUS,DIR) - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

UCB - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

Note: PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,

PRINT=(STATUS,DIR),PRINT=UCB

RESERVE= specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being erased. This is recommended during an ERASE operation to ensure that the volume will not be accessed while the erase is proceeding.

YES - the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the erase is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the erase will fail immediately.

NO - no RESERVE is issued.

The default is YES.

Warning: if RESERVE=YES is in effect, you must be sure that no other system (MVS or SCSI) has the Open System volume reserved; otherwise the FDRSOS erase will fail.

210.06 FDRSOS LABEL STATEMENT

LABEL **TYPE=SOS**

,PRINT=(STATUS,DIR)
 ,PRINT=UCB

**LABEL
STATEMENT**

This statement requests a volume labeling operation. It must be the first statement in the input; only one LABEL statement is allowed per execution. LABEL must be followed by one or more MOUNT statements with the UNIT= and SETVOL= operands to specify the Open System volumes to be labeled.

The LABEL function must be executed against each Open System volume before it can be used by other FDRSOS functions. It assigns a volume serial to a volume and records that serial in an area of the volume reserved by EMC for FDRSOS use. You only need to execute LABEL against each Open System volume when:

- the Open System volume has not previously been used by FDRSOS
- you need to change an Open System volume serial
- the EMC hardware has been reconfigured changing the size or location of the Open System volumes so that the original volume serials have been lost
- a new EMC Symmetrix subsystem is replacing the original system (such as a replacement subsystem at a disaster site).

LABEL also stores the volser in the UCB of the Open System device so that it can be used in a DISKx DD statement in other FDRSOS steps. However, that volser is lost when your MVS system is reIPLed; after an IPL, the FDRSOS VARYON function ([Section 210.07](#)) must be executed before any FDRSOS backup, restore or print is performed.

The LABEL function is also useful as a diagnostic tool, to verify that your hardware and software configuration is correct so that the Open System volume can successfully be accessed by FDRSOS. Compared to other FDRSOS functions, LABEL will give additional diagnostic messages if it cannot access the volume. In fact, if a LABEL statement is used with a MOUNT statement with only UNIT= on it (no SETVOL= operand), it will validate access to the volume but will not change its volume serial.

OPERANDS**TYPE=SOS**

Must be specified on the LABEL Statement.

PRINT=

STATUS - requests that FDRSOS attempt to identify the type of Open System volume being labeled. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in [Section 200.04](#).

(STATUS,DIR) - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

UCB - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

Note: PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,

PRINT=(STATUS,DIR),PRINT=UCB

CONTINUED. . .

210.07 FDRSOS VARYON/VARYOFF STATEMENT

VARYON	TYPE=SOS
VARYOFF	
VARYPATH	,PRINT=(STATUS,DIR)
	,PRINT=UCB

**VARYON,
VARYOFF
AND
VARYPATH
STATE-
MENTS**

These statements request a FDRSOS vary online or vary offline operation (this is not the same as a MVS VARY console command). It must be the first statement in the input; only one of these statements is allowed per execution. VARYON/VARYOFF/VARYPATH must be followed by one or more MOUNT statements with the UNIT=operand to specify the Open System devices to be varied (for VARYOFF only, the VOL= operand can be used in place of UNIT=).

The volume serial assigned to an Open System volume by the LABEL function of FDRSOS ([Section 210.06](#)) will be lost from the UCB of the device when your MVS system is reIPLed; after an IPL, the VARYON function must be executed before any FDRSOS backup, restore or print is performed. VARYON reads the stored volser from each Open System volume identified by the MOUNT statements and restores it in the UCB. Innovation recommends that you execute the VARYON function as an automatically executed task after IPL, or as the first step in any FDRSOS job.

If necessary, VARYOFF may be executed to remove the volser from the UCBs of the selected Open System volumes, making them unusable for other FDRSOS functions until another VARYON is executed. The volser will still be stored on the volume itself after a VARYOFF.

The VARYPATH statement is the same as a VARYON, except that it also insures that all channel paths (CHPIDs) assigned to the referenced Open System volumes are enabled. This might be needed if the MVS console DS PATH command, documented in [Sections 200.03](#) and [220.15](#), shows a status of * (logically off, physically on) for one or more CHPIDs for one or more Open System volumes. This might be the result of:

- A system error which set a path offline
- A MVS command of "VARY PATH(...),OFFLINE" for a Open System volume. The equivalent "VARY PATH(...),ONLINE" will fail, for the same reason that a "VARY ...,ONLINE" will fail on Open System volumes.

VARYPATH also resets some UCB flags which might prevent successful I/O; if FDRSOS jobs hang or get unexplained errors, this may solve the problem.

WARNING: Do not use the VARYPATH command unless you have the problems described above. You must execute VARYPATH only against devices which are known to be Open System volumes. The MOUNT statement for a VARYPATH command must specify only one complete unit address; multiple steps are required to VARYPATH for more than one device.

210.07 CONTINUED. . .

**VARYON,
VARYOFF
AND
VARYPATH STATE-
MENTS
(continued)**

Note: the FDRSOS VARYON and VARYOFF functions are not the same as a MVS VARY console command. The Open System device will remain offline to MVS even after execution of a FDRSOS VARYON, since MVS does not natively support FBA (Fixed Block Architecture) disks like the EMC Open System disks. However, after a VARYON, the volume serial that VARYON (and LABEL) store in the UCB of the disk may be displayed by certain MVS commands (such as the console D U command) and other utilities.

Warning: if the VARYON statement is executed with a MOUNT UNIT=* (or some other UNIT value that might include devices which are not EMC Open System disks) this might result in I/O error messages on the MVS console. FDRSOS will only attempt to VARYON devices which are generated as offline DASD devices, and uses techniques to suppress errors that result from attempting to access non-Open System devices, but in some cases this suppression may not be successful. If such error messages occur, you may be able to avoid them only by changing the UNIT= operands to include only known Open System volumes.

OPERANDS**TYPE=SOS**

Must be specified on the VARYON/VARYOFF Statement.

PRINT=

STATUS - requests that FDRSOS attempt to identify the type of Open System volume being varied. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in [Section 200.04](#).

(STATUS,DIR) - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

UCB - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

Note: PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,

PRINT=(STATUS,DIR),PRINT=UCB

CONTINUED. . .

210.08 FDRSOS MOUNT STATEMENT

MOUNT	BACKUPDSN= dsname	(RESTORE only)
M	,SETVOL= volser	(LABEL only)
	,TAPEDD= x	(DUMP/RESTORE/ERASE/PRINT only)
	,UNIT= unit	(LABEL/VARYON/VARYOFF/VARYPATH only)
	,VOL= volser	(DUMP/RESTORE/ERASE/PRINT/VARYOFF only)

**MOUNT
STATEMENT**

The MOUNT statement identifies Open System volumes to be processed by FDRSOS . For DUMP, RESTORE, ERASE and PRINT operations, either MOUNT statements or DISKx DD statements can be used to identify the volumes. For LABEL, VARYON, VARYOFF and VARYPATH operations, one or more MOUNT statements are required.

Please review the examples in [Sections 210.20-23](#) for guidance on the correct usage of the various operands of MOUNT.

OPERANDS

BACKUPDSN= For RESTORE only, optionally specifies the name of a cataloged backup data set. If the data set is a GDG, a relative generation number may be included, e.g., BACKUPDSN=BACK.OPEN1(-1). The data set will be dynamically allocated by FDRSOS and it will be restored to the Open System volume specified by VOL=; in this case, no TAPEX DD statement is required. If BACKUPDSN= is omitted, then TAPEDD= and a TAPEX DD statement must be specified (unless a DISKx DD is provided, in which case the MOUNT statement is not required).

SETVOL= Used only for LABEL operations, specifies the volume serial to be assigned to the Open System device identified by the UNIT= operand.

TAPEDD= If MOUNT statements are used with DUMP, RESTORE, ERASE, or PRINT, and the FDRSOS JCL contains more than one TAPEX DD statement, this operand is required to specify which TAPEX DD is to be used for the Open System volume identified by this MOUNT statement:

- For DUMPs, it specifies the output backup data set. If omitted, TAPE1 is assumed.
- For RESTOREs, it specifies the input backup to be restored. Either TAPEDD= with a TAPEX DD statement or the BACKUPDSN= operand on MOUNT may be used.
- For ERASEs, TAPEDD= is required only if MAXTASKS=n is specified on the ERASE statement; there must be a TAPEX DD DUMMY and a matching SYSPRINx DD specifying the destination for messages for this ERASE operation.
- For PRINTs, it specifies the output print data set. If omitted, TAPE1 is assumed.

The operand is the single character "x", e.g., TAPEDD=3 specifies TAPE3. A unique value must be used on each MOUNT statement since each Open System volume must be directed to a different TAPEX DD.

210.08 CONTINUED. . .

UNIT=

Specifies the S/390 device address(s) of Open System volume(s) to be processed for LABEL, VARYON, VARYOFF and VARYPATH operations (it is not used with DUMP, RESTORE, ERASE or PRINT). For VARYOFF only, the VOL= may alternately be used to identify the volume.

For LABEL, the value is a 4-digit S/390 device address, e.g., UNIT=01E0 or UNIT=25C6. The device at the address given will be validated as an EMC Open System volume and will be labeled with the value specified by the SETVOL= operand.

For VARYON and VARYOFF, the value may be a 4-digit S/390 device address (e.g., UNIT=01E0) or may be a prefix followed by an asterisk, e.g., UNIT=01E* or UNIT=03*. You may also specify a list of units and/or prefixes by enclosing the list in parenthesis, e.g., UNIT=(01E*,01F*,01D5). You may even use UNIT=* to specify all DASD device addresses in your system. All device addresses in the I/O configuration matching the address or prefix will be validated as EMC Open System devices (offline to MVS and responding to FBA commands); for those that pass, FDRSOS will read the volume serial recorded on the volume by the LABEL function and store it in the MVS UCB for the device.

For VARYPATH, the value is a 4-digit S/390 device address, e.g., UNIT=01E0 or UNIT=25C6. All defined paths (CHPIDs) for the device are enabled (unless they are physically offline) and certain UCB flags that may inhibit I/O are reset.

UNIT= may be used with ERASE operations, if the volume to be erased has not been previously assigned a pseudo volume serial with a LABEL statement. However, Innovation recommends labeling the volume and using the VOL= parameter to identify the volume to be erased.

WARNING: 4-digit device addresses must be used even if your operating system only supports 3-digit addresses for other functions; specify a leading zero if required.

VOL=

Specifies the volume serial of the Open System volume to be processed by DUMP, RESTORE, ERASE, PRINT, or VARYOFF. For VARYOFF only, a volume serial prefix may be specified by following it with an asterisk (e.g., VOL=OPEN*). A volume serial can be used to identify an Open System volume only if a FDRSOS LABEL or VARYON operation has been run since the LAST IPL to store the volume serial in the MVS UCB.

For VARYOFF only, a volume serial prefix may be specifying by following it with an asterisk, e.g., VOL=OPEN*. You can also specify a list of volsers and/or prefixes by enclosing the list in parenthesis, e.g., VOL=(OPEN*,EMC123,NTA1*).

For VARYOFF, either UNIT= or VOL= or both must be specified. For DUMP, RESTORE, or PRINT VOL= must be given. For LABEL and VARYON, UNIT= is required.

CONTINUED. . .

210.09 FDRSOS SELECT/EXCLUDE STATEMENT

```

SELECT          FROMBLK=nnnnn,TOBLK=nnnnn
S
                ,TAPEDD=x

EXCLUDE        ,VOL=volser
X

```

**SELECT AND
EXCLUDE
STATE-
MENTS**

The SELECT and/or EXCLUDE statements are required when TYPE=PARTIAL was specified on the DUMP, RESTORE, or PRINT statement to specify the data to be dumped, restored, or printed.

SELECT identifies a range of data blocks to be processed and EXCLUDE identifies data blocks from within those selected by SELECT statements which are not to be processed. Multiple SELECT and EXCLUDE statements may be input to specify multiple ranges.

Open System volumes are formatted as a series of 512 byte data sectors, but are usually read and written as 4K (4096) byte blocks, so the SELECT and EXCLUDE statements specify a range of 4K blocks by block number. Block numbers are relative to zero, so 0 will be the first 4K block on the volume, 1 the second 4K block, etc.

The control statements are always scanned in the order in which they were input, so in general, EXCLUDE statements should precede SELECT statements.

OPERANDS

FROMBLK= Specifies the beginning and end of a range of 4K data blocks to be processed for backup, restore, or print. The values for FROMBLK= and TOBLK= are block numbers relative to zero. In other words, FROMBLK=0,TOBLK=499 will select the first 500 4K blocks on the volume. TOBLK= must be equal or greater than FROMBLK=.

TOBLK=

TAPEDD= If the FDRSOS JCL contains more than one TAPEX DD statement, TAPEDD= may be used to specify to which TAPEX DD this SELECT/EXCLUDE applies. The operand is the single character "x", e.g., TAPEDD=3 specifies TAPE3.

VOL= If more than one Open System volume is being processed in this step, VOL= specifies the volume serial of the Open System volume to which this SELECT/EXCLUDE applies. It may specify a complete volser (e.g., VOL=OPEN23) or a volser prefix (followed by an asterisk, e.g., VOL=OPEN*)

Either TAPEDD= or VOL= may be specified to identify the volume to which this statement applied. If both are omitted, this statement will apply to all Open System volumes being processed in this step.

210.10 FDRSOS PRINT MODIFIERS

FDRSOS includes the ability to scan for specified ASCII or hexadecimal strings of data while printing data from Open System volumes (the PRINT statement is described in [Section 210.04](#)). Only the data sectors containing the specified string will be printed. If scan statements are omitted, all selected data sectors are printed.

**\$TRKPRIN
DD
STATEMENT**

In order to specify scan statements, an additional DD statement must be added to the FDRSOS step. \$TRKPRIN contains the scan statements and is usually a DD * (input stream) data set, separate from the normal SYSIN DD statement. The input may include a DEFAULT statement and/or a SCAN statement.

**DEFAULT
STATEMENT**

DEFAULTCHECKSUM | **NOCHECKSUM**

,LINECOUNT=nnn

,NOZEROBLOCKS

,SEP | NOSEP

The DEFAULT statement is optional, but must come first if present. It specifies options for the PRINT operation and can be present even if the SCAN statement is omitted to control non-scan PRINT functions.

OPERANDS**CHECKSUM**

CHECKSUM specifies that a hexadecimal checksum be calculated

NOCHECKSUM

and printed in every separator line (see SEP below). The checksum is calculated by adding every 4-byte word in the data; the value is a running checksum, meaning that the value printed at any point is the sum of all 4-byte words printed since the beginning of the PRINT operation. This checksum might be useful for identifying sectors which have changed, or for quickly verifying that a FDRSOS restore has restored the exact same data that existed before the backup. If the last checksum printed in 2 PRINT operations is the same, the data printed is probably the same (it is possible to have offsetting changes in data that result in the same checksum, but this is usually not the case). **NOCHECKSUM** suppresses the checksum calculation.

The default is CHECKSUM.

LINECOUNT=

specifies the lines per page to be printed on the TAPEX DD statement. The default is 58.

NOZEROBLOCKS

If specified, sectors which are entirely binary zeros will not be printed. This is useful to restrict the printout to just sectors containing significant information.

SEP**NOSEP**

SEP indicates that every 512 data sector printed will have a separator line, identifying the sector number and, if requested, displaying the current checksum including that sector.

NOSEP requests that the separator line be printed only at the beginning of each group of 64 data sectors, identifying the range of sectors and displaying the current checksum through the last sector in the range.

210.10 CONTINUED . . .

**SCAN
STATEMENT**

SCAN **ARGUMENT**=string | 'string' | X'hexdata'
 ,CASE=UPPER | **LOWER**

The SCAN statement is optional. If present, it specifies that only data sectors that contain a specified ASCII or hexadecimal string should be printed.

OPERANDS

ARGUMENT= specifies the string which should scanned. It can be in one of 3 formats:
string - is an alphanumeric ASCII character string. It cannot contain blanks or special characters.
'string' - is an ASCII string in quotes. It can contain any valid ASCII characters including blanks.
X'hexdata' - is a string of data in hexadecimal. It must contain an even number of valid hexadecimal digits (0-9, A-F).

Note: ASCII strings are actually entered on the SCAN statement in EBCDIC; they are translated to ASCII for use in the scan.

CASE= If an ASCII string was specified for ARGUMENT=, CASE specifies translation of that string.
UPPER - the string is translated to upper case ASCII
LOWER - the string is translated to lower case ASCII

The default is UPPER.

Note: mixed-case ASCII strings can be entered only by entering the hexadecimal equivalent of the ASCII characters using the ARGUMENT=X'hexdata' option.

210.20 FDRSOS INITIALIZATION EXAMPLES

**LABEL
OPEN
SYSTEM
VOLUMES**

Assign volume serials to a number of Open System volumes. The volser specified is stored in an area of the volume reserved by EMC for FDRSOS use and is also stored in the UCB (MVS Unit Control Block) of the device, allowing it to be referenced in JCL by other FDRSOS jobs. The LABEL function needs to be executed only once for a given Open System volume unless you need to change the label or the label is lost because of EMC hardware reconfiguration or replacement. PRINT=STATUS requests that FDRSOS identify the type and contents of the Open System volume, to document that you are labeling the correct volume. LABEL will also provide additional diagnostics if it cannot access the specified unit as an Open System volume.

```
//LABEL      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//SYSIN      DD        *
              LABEL     TYPE=SOS,PRINT=STATUS
              MOUNT     UNIT=01E0,SETVOL=OPEN#1
              MOUNT     UNIT=01E1,SETVOL=OPEN#2
```

**VERIFY
ACCESS TO
SYSTEM
VOLUME**

The LABEL function can be used with no SETVOL= operand on the MOUNT statement, as shown, to verify that the specified device is an Open System volume in an EMC Symmetrix subsystem, and that the device can be successfully accessed by FDRSOS, but it will not change the volume serial of the volume. This can be used to verify that your hardware and software configuration is correct for use by FDRSOS; use the example above for the same purpose if you are ready to label the Open System volumes. PRINT=UCB will format the MVS Unit Control Block of the selected device. If the device cannot be accessed, additional diagnostic messages are printed to help identify the problem ([See Section 220.15](#) for diagnostic help). Note that if the LABEL function gets message FDR212 REASON=A SETVOL MISSING, the test was successful.

```
//ACCESS     EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//SYSIN      DD        *
              LABEL     TYPE=SOS,PRINT=UCB
              MOUNT     UNIT=01E0
```

CONTINUED. . .

210.20 CONTINUED . . .

**VARYON
OPEN
SYSTEM
VOLUMES**

The volume serial of an Open System volume stored in the UCB (MVS Unit Control Block) is not preserved across IPLs. The VARYON function reads the volser from the FDRSOS area of the volume and stores it in the UCB again, allowing it to be referenced in JCL by other FDRSOS jobs. This example executes the VARYON for two ranges of Open System Devices (01E0-01EF, 01F0-01FF). The VARYON function is not the same as a MVS VARY console command; the devices will remain offline to MVS. PRINT=(STATUS,DIR) requests that FDRSOS identify the type and contents of each Open System volume processed; for some volume formats the contents of the root directory is also displayed.

```
//VARYON      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD      DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//SYSIN       DD      *
               VARYON  TYPE=SOS,PRINT=(STATUS,DIR)
               MOUNT   UNIT=01E*
               MOUNT   UNIT=01F*
               MOUNT   UNIT=01F*
```

**VARYON
ALL OPEN
SYSTEM
VOLUMES**

The volume serial of an Open System volume stored in the UCB (MVS Unit Control Block) is not preserved across IPLs. The VARYON function reads the volser from the FDRSOS area of the volume and stores it in the UCB again, allowing it to be referenced in JCL by other FDRSOS jobs. This example executes the VARYON for several ranges of addresses (only offline DASD devices are examined). Since the FDRSOS control statements are completely specified in the PARM data, this example could be converted into a PROC for a started task to be executed automatically at IPL time.

Warning: VARYON with MOUNT UNIT=* might result in I/O error messages on the MVS console. FDRSOS will only attempt to VARYON devices which are generated as offline DASD devices, and uses techniques to suppress errors that result from attempting to access non-Open System devices, but in some cases this suppression may not be successful. If such error messages occur, you may be able to avoid them by changing the UNIT= operands to include only known Open System volumes.

```
//VARYON      EXEC    PGM=FDRSOS,REGION=0M,
//      PARM=' VARYON TYPE=SOS/MOUNT UNIT=(01E*,01F*,03C*,03D*) '
//STEPLIB     DD      DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//SYSIN       DD      DUMMY
```

CONTINUED. . .

210.20 CONTINUED . . .

**VARYOFF
OPEN
SYSTEM
VOLUMES**

You might want to make certain Open System volumes unavailable to FDRSOS. The VARYOFF function removes the volser of specified Open System volumes from the MVS UCB. A subsequent VARYON would be required to make them available again. As shown, the volumes involved can be specified by unit address or volser. The VARYOFF function is not the same as a MVS VARY console command; Open System devices are always offline to MVS.

```
//VARYOFF EXEC PGM=FDRSOS,REGION=0M
//STEPLIB DD DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
        VARYOFF TYPE=SOS
        MOUNT UNIT=01E*
        MOUNT VOL=OPEN2*
```

**VARY PATHS
FOR AN
OPEN SYS-
TEM
VOLUME**

When some paths (CHPIDs) are not available for an Open System volume, or I/O to a volume is not successful for an undetermined reason, the VARYPATH command may be able to correct the problem. It resets some flags which may prevent successful I/O, and enables all paths less they are physically varied offline. The MVS console command DSPATH (see Section 220.15) will show path status; a status of * for a CHPID indicates it is logically offline but physically online. The MOUNT statement must specify only a single Open System device.

```
//VARYPATH EXEC PGM=FDRSOS,REGION=0M
//STEPLIB DD DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
        VARYPATH TYPE=SOS
        MOUNT UNIT=01E0
```

210.21 FDRSOS BACKUP EXAMPLES

**DUMP ONE
VOLUME**

Dump an Open System disk volume to 3490E tape cartridges, creating 2 copies. The TAPE11 DD can be omitted if only one copy is required. A MOUNT statement is used to specify the volume to be backed up. A volume count of 99 is specified on the TAPE DD statements in case more than 5 tape volumes are required.

```
//DUMP      EXEC PGM=FDRSOS,REGION=0M
//STEPLIB   DD  DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT  DD  SYSOUT=*
//FDRSUMM   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//TAPE1     DD  UNIT=3490,DSN=BACKUP.VE#01E0.C1,DISP=(,CATLG),VOL=(,,99)
//TAPE11    DD  UNIT=3490,DSN=BACKUP.VE#01E0.C2,DISP=(,CATLG),VOL=(,,99)
//SYSIN     DD  *
              DUMP  TYPE=FULL
              MOUNT VOL=E#01E0
```

The report on SYSPRINT will look similar to:

FDR007	STARTING TIME OF FULL VOL DUMP -- 16.18.11 -- UNIT=01E0	,IN=D#E#01E0,OUTPUT=TAPE1
FDR007	ENDING TIME OF FULL VOL DUMP -- 16.30.08 -- UNIT=01E0	,IN=D#E#01E0,OUTPUT=TAPE1
FDR122	OPERATION STATISTICS FOR SOSVOLUME.E#01E0	
FDR122	BYTES ON VOLUME	4,355,850,240
FDR122	DATASETS PROCESSED	0
FDR122	BYTES READ FROM DASD	4,355,850,240
FDR122	BYTES ON BACKUP	4,364,783,536
FDR122	COMPRESSION SAVINGS (%)	0
FDR122	DASD SECTORS BACKED UP	8,507,520
FDR122	BACKUP BLOCKS WRITTEN	79,760
FDR122	DASD EXCPS	8,863
FDR122	BACKUP FILE EXCPS	8,864
FDR122	CPU TIME (SECONDS)	14.040
FDR122	ELAPSED TIME (MINUTES)	12.4
FDR122	BACKUP TIME(EXCLUDING MOUNTS)	12.0
FDR122	BACKUP COPY 1 ON TAPE DSN=BACKUP.VE#01E0.C1	
FDR122	VOL=900013,900014,900015	
FDR122	BACKUP COPY 2 ON TAPE DSN=BACKUP.VE#01E0.C2	
FDR122	VOL=900022,900023,900024	
FDR002	FDR DUMP SUCCESSFULLY COMPLETED VOL=E#01E0	

and the report on FDRSUMM will be similar to:

VOLSER	COMP CODE	ELAPSED TIME(MIN)	VOLUME SIZE	DASD BYTES READ FROM VOL	BYTES ON BACKUP FILE	COMP- PRESS	SECTORS DUMPED
E#01E0	0	12.0	4,355,850,240	4,355,850,240	4,364,783,536	0%	8,507,520

Reports from other dump and restore operations in these examples will also be similar.

210.21 CONTINUED . . .

**DUMP
MULTIPLE
VOLUMES
SERIALLY**

Dump three Open System disk volumes sequentially to 3590 (Magstar) tape cartridges, using IDRC hardware compression. The 3 backups will be stacked on the tape as consecutive tape data sets. DISKx DD statements are used to specify the volumes to be backed up; each DISKx will be backed up to the corresponding TAPEx (e.g., DISK3 to TAPE3). OPENSYS must be a esoteric unit name in your I/O configuration which includes the EMC Open System device addresses; if such a name does not exist, you must use UNIT=3390 (or 3380 depending on how the devices were defined).

```
//DUMP      EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB   DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT  DD    SYSOUT=*
//FDRSUMM   DD    SYSOUT=*
//SYSUDUMP  DD    SYSOUT=*
//DISK3     DD    UNIT=OPENSYS,VOL=SER=OPEN#1,DISP=OLD
//DISK5     DD    UNIT=OPENSYS,VOL=SER=OPEN#2,DISP=OLD
//DISK8     DD    UNIT=OPENSYS,VOL=SER=OPEN#3,DISP=OLD
//TAPE3     DD    DSN=BACKUP.OPEN1,DISP=(,CATLG),UNIT=3590,DCB=TRTCH=COMP
//TAPE5     DD    DSN=BACKUP.OPEN2,DISP=(,CATLG),VOL=REF=*.TAPE3,LABEL=2,
//          DCB=TRTCH=COMP
//TAPE8     DD    DSN=BACKUP.OPEN3,DISP=(,CATLG),VOL=REF=*.TAPE5,LABEL=3,
//          DCB=TRTCH=COMP
//SYSIN     DD    *
DUMP      TYPE=FULL
```

**DUMP
MULTIPLE
VOLUMES
CONCUR-
RENTLY**

Dump three Open System disk volumes concurrently to 3 different tape drives. The backup data sets are GDGs. MOUNT statements are used to specify which volumes to backup and to which TAPEx DD they are written. A job like this might be used to backup volumes which must be backed up at the same time, such as members of a AIX/6000 volume group. PRINT=STATUS requests that FDRSOS identify the type and contents of each Open System volume dumped, to document that the correct volumes were backed up.

```
//DUMP      EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB   DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT  DD    SYSOUT=*
//SYSPRINA  DD    SYSOUT=*
//SYSPRINB  DD    SYSOUT=*
//SYSPRINC  DD    SYSOUT=*
//FDRSUMM   DD    SYSOUT=*
//SYSUDUMP  DD    SYSOUT=*
//TAPEA     DD    DSN=BACKUP.OPEN1(+1),DISP=(,CATLG),UNIT=TAPE,VOL=(,,99)
//TAPEB     DD    DSN=BACKUP.OPEN2(+1),DISP=(,CATLG),UNIT=TAPE,VOL=(,,99)
//TAPEC     DD    DSN=BACKUP.OPEN3(+1),DISP=(,CATLG),UNIT=TAPE,VOL=(,,99)
//SYSIN     DD    *
DUMP      TYPE=FULL,MAXTASKS=3,PRINT=STATUS
MOUNT     VOL=EMC120,TAPEDD=A
MOUNT     VOL=EMC123,TAPEDD=B
MOUNT     VOL=EMC128,TAPEDD=C
```

Note: FDRSOS software compression (invoked by the COMPRESS= operand on DUMP statements) is recommended only when the output backup data set is on disk or is on tapes connected by slower parallel (bus/tag) channels, such as 3480 tape drives. For disk output it reduces the backup data set size, while on parallel-attached tapes it reduces channel utilization, allowing for more concurrent backups. For tapes connected via ESCON (serial) channels, the channel is fast enough that channel utilization is not a problem, so hardware IDRC compression is recommended.

210.21 CONTINUED . . .

**DUMP
SELECTED
DATA
BLOCKS**

Selected ranges of data blocks are dumped from an Open System volume to a backup data set on disk. FDRSOS compression is used to reduce the size of the backup (this is recommended when outputting to disk backup data sets). Since it is difficult to tell what data resides in specific blocks on an Open System volume, this procedure will not be used for normal operations.

```
//DUMP          EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB       DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT      DD        SYSOUT=*
//FDRSUMM       DD        SYSOUT=*
//SYSUDUMP      DD        SYSOUT=*
//TAPE1         DD        DSN=TECH.BACKUP.OPEN1,UNIT=SY-
DA,SPACE=(CYL,(20,5),RLSE),
//              DISP=(NEW,CATLG)
//DISK1         DD        UNIT=3390,VOL=SER=AIX123,DISP=OLD
//SYSIN         DD        *
                DUMP      TYPE=PARTIAL,COMPRESS=ALL
                SELECT    FROMBLK=0,TOBLK=499
                SELECT    FROMBLK=1050,TOBLK=1070
```


210.22 FDRSOS RESTORE EXAMPLES**FULL
VOLUME
RESTORE**

Restore an Open System volume. By default, the operator will be prompted for permission before beginning the restore.

```
//RESTORE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB      DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD        SYSOUT=*
//FDRSUMM      DD        SYSOUT=*
//SYSUDUMP     DD        SYSOUT=*
//TAPE1        DD        DSN=BACKUP.VOP3132.C1,DISP=OLD
//DISK1        DD        UNIT=3390,VOL=SER=OP3132,DISP=OLD
//SYSIN        DD        *
RESTORE        TYPE=FULL
```

**SERIAL
VOLUME
RESTORE**

Restore several Open System volumes serially. CONFMESS=NO indicates that the operator will not be prompted for permission before beginning the restores. Each of the indicated backup data sets will be dynamically allocated and restored to the indicated Open System volume. This might be used to restore volumes which must be processed together, such as members of an AIX/6000 volume group. Because BACKUPDSN= is used to specify the backups, the restores will be done serially, one at a time. PRINT=(STATUS,DIR) requests that FDRSOS identify the type and contents of each Open System volume both before and after the restore; for some volume types the contents of the root directory is also displayed. This will help document that the proper backup has been restored.

```
//RESTORE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB      DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD        SYSOUT=*
//FDRSUMM      DD        SYSOUT=*
//SYSUDUMP     DD        SYSOUT=*
//SYSIN        DD        *
RESTORE        TYPE=FULL,CONFMESS=NO,PRINT=(STATUS,DIR)
MOUNT          VOL=EMC120,BACKUPDSN=BACKUP.OPEN1(0)
MOUNT          VOL=EMC123,BACKUPDSN=BACKUP.OPEN2(0)
MOUNT          VOL=EMC128,BACKUPDSN=BACKUP.OPEN3(0)
```

**CON-
CURRENT
VOLUME
RESTORE**

Restore several Open System volumes concurrently. By default, the operator **will** be prompted for permission before beginning the restores. Each TAPE_x backup will be restored to the corresponding DISK_x volume. Messages from each restore are printed on the SYSPRIN_x DD. This might be used to restore volumes which must be processed together, such as members of an AIX/6000 volume group.

```
//RESTORE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB      DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD        SYSOUT=*
//FDRSUMM      DD        SYSOUT=*
//SYSUDUMP     DD        SYSOUT=*
//TAPE1        DD        DSN=BACKUP.VAIX01A,DISP=OLD
//DISK1        DD        UNIT=3390,VOL=SER=AIX01A,DISP=OLD
//SYSPRIN1     DD        SYSOUT=*
//TAPE2        DD        DSN=BACKUP.VAIX01B,DISP=OLD
//DISK2        DD        UNIT=3390,VOL=SER=AIX01B,DISP=OLD
//SYSPRIN2     DD        SYSOUT=*
//SYSIN        DD        *
RESTORE        TYPE=FULL,MAXTASKS=2
```

210.22 CONTINUED. . .

**SELECTED
BLOCK
RESTORE**

Restore selected data blocks to an Open System volume. The backup must contain the indicated data blocks; it might be a full backup of the volume, or a selected block backup that includes those blocks. The blocks will always be restored to their original locations on the disk; there is no facility for restoring to a different location. The backup data set is dynamically allocated.

```
//RESTORE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB      DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD        SYSOUT=*
//FDRSUMM      DD        SYSOUT=*
//SYSUDUMP     DD        SYSOUT=*
//SYSIN        DD        *
      RESTORE      TYPE=PARTIAL
      MOUNT        VOL=E#01E3,BACKUPDSN=PROD.BACKUP.OPEN3
      SELECT       FROMBLK=0,TOBLK=5
```

**ALTERNATE
VOLUME
RESTORE**

Restore several Open System volumes concurrently to alternate (spare) volumes. This technique can be used for recovery of individual files; see [Section 220.14](#) for more details. The operator will be prompted for permission before beginning the restores. Each TAPEx backup will be restored to the corresponding DISKx volume, but the FDRSOS Open System volume serial of the output volumes will be preserved, and processing will be performed to preserve the physical volume IDs of the volumes and make the volumes usable on the Open System. Messages from each restore are printed on the SYSPRINx DD. PRINT=STATUS requests that FDRSOS identify the type and contents of each Open System volume restored, to document that the correct backups were restored.

```
//RESTORE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB      DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD        SYSOUT=*
//FDRSUMM      DD        SYSOUT=*
//SYSUDUMP     DD        SYSOUT=*
//TAPE1        DD        DSN=BACKUP.VAIX01A,DISP=OLD
//DISK1        DD        UNIT=3390,VOL=SER=AIX27A,DISP=OLD
//SYSPRIN1     DD        SYSOUT=*
//TAPE2        DD        DSN=BACKUP.VAIX01B,DISP=OLD
//DISK2        DD        UNIT=3390,VOL=SER=AIX27B,DISP=OLD
//SYSPRIN2     DD        SYSOUT=*
//SYSIN        DD        *
      RESTORE      TYPE=FULL,MAXTASKS=2,CONFMESS=YES,
                  CPYVOLID=NO,VOLRESET=NO,PRINT=STATUS
```

210.23 FDRSOS PRINT EXAMPLES

**PRINT
SELECTED
BLOCKS**

Print selected blocks from an Open System Volume.

```
//PRINT      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//FDRSUMM    DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//TAPE1      DD        SYSOUT=*
//SYSIN      DD        *
PRINT        TYPE=PARTIAL
MOUNT        VOL=EMC123
SELECT       FROMBLK=0,TOBLK=49
SELECT       FROMBLK=500,TOBLK=510
```

**PRINT NON-
ZERO
BLOCKS**

Print selected blocks from an Open System Volume, but only print those blocks which are not all binary zeros. Innovation may ask you to run a PRINT job similar to this to help us understand new or unusual disk formats.

```
//PRINT      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//TAPE1      DD        SYSOUT=*
//SYSIN      DD        *
PRINT        TYPE=PARTIAL
MOUNT        VOL=SUN01X
SELECT       FROMBLK=0,TOBLK=300
//$TRKPRIN   DD        *
DEFAULT      NOZEROBLOCKS
```

**SCAN FOR
ASCII DATA**

An entire Open System volume will be scanned for a specified ASCII string. Only those data sectors containing the string will be printed.

```
//PRINT      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//FDRSUMM    DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//DISK1      DD        UNIT=3390,VOL=SER=EMC123,DISP=OLD
//TAPE1      DD        SYSOUT=*
//SYSIN      DD        *
PRINT        TYPE=FULL
//$TRKPRIN   DD        *
SCAN         ARGUMENT='test data',CASE=LOWER
```

210.23 CONTINUED. . .

**SCAN FOR
HEX DATA**

Selected data blocks on an Open System volume will be scanned for a specified hexadecimal string. Only those data sectors containing the string will be printed. No data checksums will be printed.

```
//PRINT      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//FDRSUMM    DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//TAPE1      DD        SYSOUT=*
//SYSIN      DD        *
PRINT        TYPE=PARTIAL
MOUNT        VOL=EMC123
SELECT       FROMBLK=0,TOBLK=999
//$TRKPRIN   DD        *
DEFAULT      NOCHECKSUM
SCAN         ARGUMENT=X'00134790'
```

210.24 FDRSOS ERASE EXAMPLES

WARNING: the ERASE function will overwrite all data and preformatting on an Open System volume, and must be used with great care. After an ERASE, the volume will not be usable again until it is reformatted by utilities on the Open System involved. ERASE is most likely to be useful after a disaster test or when replacing an EMC Symmetrix containing Open System Data, to insure that all corporate data is removed. Although the CONFERENCE=NO operand can be used to suppress the prompt for operator permission, Innovation recommends that you do not use it except at a disaster recovery site (and even then with care).

You should run a VARYON function with PRINT=STATUS before the ERASE to confirm that the correct volume has been selected for erasure (see example in [Section 210.20](#)).

**ERASE A
VOLUME**

The volume specified by the MOUNT statement will be erased, entirely overwritten with binary zeros. The operator will be prompted for permission before erasing the volume. PRINT=STATUS requests that FDRSOS identify the type and contents of each Open System volume before it is erased, to document the data that was erased (after the erase it will identify it as "UNKNOWN VOLUME TYPE" to confirm that the erasure took place.

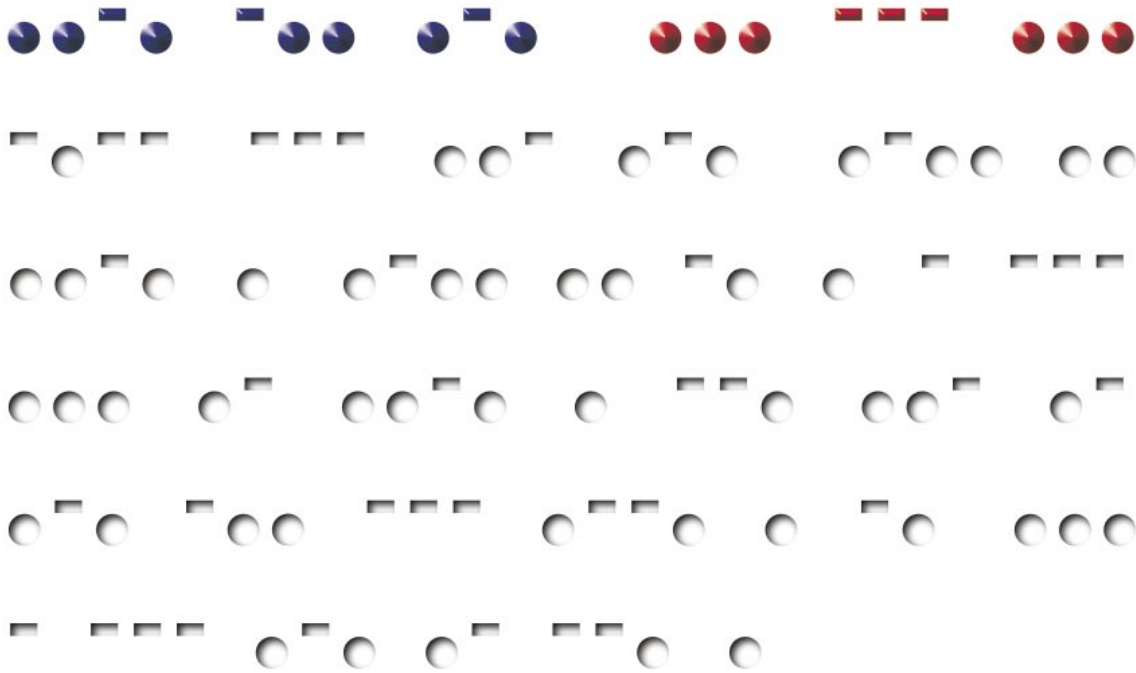
```
//ERASE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//FDRSUMM    DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//SYSIN      DD        *
ERASE        TYPE=FULL,PRINT=STATUS
MOUNT        VOL=EMC123
```

**ERASE
SEVERAL
VOLUMES
CONCUR-
RENTLY**

The volumes specified by the DISKx DD statements will be erased, entirely overwritten with binary zeros. The operator will be prompted for permission before erasing the volumes.

```
//ERASE      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD        DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD        SYSOUT=*
//FDRSUMM    DD        SYSOUT=*
//SYSUDUMP   DD        SYSOUT=*
//DISK1      DD        UNIT=3390,VOL=SER=OPEN#1,DISP=OLD
//TAPE1      DD        DUMMY
//SYSPRIN1   DD        SYSOUT=*
//DISK2      DD        UNIT=3390,VOL=SER=OPEN#2,DISP=OLD
//TAPE2      DD        DUMMY
//SYSPRIN2   DD        SYSOUT=*
//DISK3      DD        UNIT=3390,VOL=SER=OPEN#3,DISP=OLD
//TAPE3      DD        DUMMY
//SYSPRIN3   DD        SYSOUT=*
//SYSIN      DD        *
ERASE        TYPE=FULL,MAXTASKS=3
```

THIS PAGE WAS INTENTIONALLY LEFT BLANK



Installation and User's Guide



THIS PAGE WAS INTENTIONALLY LEFT BLANK

220.01 FDRSOSTC BACKUP COPY UTILITY

OVERVIEW The FDRSOS tape copy utility (FDRSOSTC) has been specifically designed to copy FDRSOS-formatted backups on tape or disk. **As noted earlier, FDRSOS backups cannot be correctly copied by any non-Innovation utilities.**

FDRSOSTC may be used to copy any FDRSOS backup, from disk or tape, to disk or tape, optionally creating a second copy at the same time. Backups in FDRSOS compressed format can be copied, and FDRSOSTC can create compressed output from uncompressed input, and vice versa. Simple JCL and control statements are used to specify the input and output backup data sets.

220.02 FDRSOSTC JCL REQUIREMENTS

FDRSOSTC requires the following JCL to execute:

**EXEC
STATEMENT**

Specifies the program name (FDRSOSTC), and region requirement (a region of 4M is adequate for all functions) and optional PARM= field which may contain the FDRSOS control statement (a COPY statement). For example,

```
//SOS EXEC PGM=FDRSOS,REGION=0M,PARM=' COPY COMPRESS=NONE '
```

**STEPLIB DD
STATEMENT**

If required, specifies the load library in which FDRSOSTC resides. It must be an APF authorized library.

**SYSPRINT DD
STATEMENT**

Specifies the output message data set. Normally a SYSOUT data set.

**SYSUDUMP
DD
STATEMENT**

Specifies the abend dump data set. Although not required, we strongly urge you to always include this DD statement, so that we can help you diagnose error conditions. Usually specifies a SYSOUT data set.

**TAPEIN DD
STATEMENT**

Specifies the input tape or disk FDRSOS backup data set. At least DSN= and DISP= must be given; if the input data set is not cataloged then UNIT=, VOL=, and possibly LABEL= must also be given.

**TAPEOUT DD
STATEMENT**

Specifies the primary output tape or disk to be created.

If more than 5 output tape volumes may be required, you must specify a volume count in the VOL= parameter, e.g., "VOL=(,,,99)". Innovation recommends always specifying a volume count when the output is on tape.

You may want to specify the RETPD= or EXPDT= JCL parameters to specify the expiration of the output file.

If the output is on disk, the DD statement can create the output file (with DISP=(NEW,KEEP) or (NEW,CATLG) and SPACE=) or can refer to an existing output file (DISP=OLD).

TAPEOUT may be DUMMY; TAPEIN will still be read. This is useful to verify the readability of the input files, since FDRSOSTC will validate the contents and format of the backup file (Note: if the input file is compressed, specify COMPRESS=NONE for full validation).

**TAPE2OUT
DD
STATEMENT**

(Optional) Specifies that a second output copy is to be created. All comments about 'TAPEOUT' above apply to 'TAPE2OUT'. If a TAPEOUT2 DD is present, it will be treated as an alias for TAPE2OUT. If the TAPE2OUT and TAPEOUT2 DDs are omitted, then the TAPEOUT file will be the only copy created.

**SYSIN DD
STATEMENT**

Specifies the control statement data set. Usually an input stream or DD * data set. It can be omitted (or specified as DUMMY); if so, and no PARM= was specified on the EXEC statement, a COPY statement with no operands will be assumed.

220.03 FDRSOSTC CONTROL STATEMENTS

COPY **BUFNO=**MAX | nn
,COMPRESS=ALL | COPY1 | COPY2 | NONE

**COPY
STATEMENT**

The COPY control statement is the only statement accepted by FDRSOSTC. All operands are optional. If the COPY statement itself is omitted, the defaults for all operands are assumed. So, unless you need to override one of the operands, you can execute most FDRSOSTC steps with no control statements at all (in which case the SYSIN DD statement can be omitted or specified as DUMMY).

OPERANDS**BUFNO=**

specifies how many buffers will be used while copying each Open System backup. Each buffer is 36K in length. The buffers acquired will be divided into 2 sets in order to overlap input and output I/O operations; each I/O will read or write one half of the buffers. Reducing the number of buffers will reduce the amount of below-the-line storage required while copying, but will also reduce the efficiency of the copy and increase the elapsed time.

MAX - specifies that 32 buffers will be acquired.

nn - the specified number of buffers is acquired. The value may be from 1 to 32 but it will be rounded up to the next higher even number. Values over 32 are treated as 32.

The default is MAX (32).

COMPRESS=

ALL specifies the backup copy on TAPEOUT and TAPE2OUT (if present) are both to be compressed using FDRSOS software compression.

COPY1 specifies that only the TAPEOUT backup is to be compressed. TAPE2OUT (if present) will be uncompressed.

COPY2 specifies that only the TAPE2OUT backup (if present) is to be compressed. TAPEOUT will be uncompressed.

NONE specifies that neither backup will be compressed.

Compression of the output backups is independant of whether the input backup was compressed or not. The input data will be uncompressed or compressed as appropriate to create the output.

The default is ALL if the backup on TAPEIN was compressed, or NONE if it was not compressed.

220.04 FDRSOSTC EXAMPLES

**COPY ONE
BACKUP**

One Open System backup on tape or disk is to be copied to another tape. The input data set is cataloged and the output data set will be cataloged. Since no SYSIN DD statement is present, the defaults for all operands of the COPY statement are assumed. By default, if the input backup was compressed by FDRSOS software (the COMPRESS= option on the DUMP statement) the output backup will also be compressed; otherwise, it will not be compressed.

```
//COPY          EXEC  PGM=FDRSOSTC,REGION=4M
//STEPLIB       DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT      DD    SYSOUT=*
//SYSUDUMP      DD    SYSOUT=*
//TAPEIN        DD    DSN=BACKUP.OPEN1,DISP=OLD
//TAPEOUT       DD    DSN=BACKUP.OPEN1.COPY2,DISP=(,CATLG),UNIT=CART
```

**COPY
BACKUP
CREATING
2 COPIES**

One Open System backup on tape is to be copied creating 2 identical outputs on tape. The input data set is cataloged and the output data sets will be cataloged. The outputs will not be compressed by FDRSOS software compression (COMPRESS=NONE) but will be compressed by IDRC hardware compression (TRTCH=COMP).

```
//COPY          EXEC  PGM=FDRSOSTC,REGION=4M
//STEPLIB       DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT      DD    SYSOUT=*
//SYSUDUMP      DD    SYSOUT=*
//TAPEIN        DD    DSN=BACKUP.OPEN1,UNIT=3490,VOL=SER=B01234,DISP=OLD
//TAPEOUT       DD    DSN=BACKUP.OPEN1.COPY2,DISP=(,CATLG),
//              UNIT=3490,DCB=TRTCH=COMP
//TAPE2OUT      DD    DSN=BACKUP.OPEN1.COPY3,DISP=(,CATLG),
//              UNIT=3490,DCB=TRTCH=COMP
//SYSIN         DD    *
                COPY  COMPRESS=NONE
```

CONTINUED...

220.04 CONTINUED . . .

**COPY
MULTIPLE
BACKUPS**

Three Open System backups are to be copied to 3 consecutive files on the output tape. The input backup files are cataloged, but if they happen to be files on the same input tape, the RETAIN parameter insures that the input tape is not dismounted between steps. The COPY statement is specified via the PARM= on the EXEC statement, so no SYSIN DD statements are required.

```
//COPY1      EXEC  PGM=FDRSOSTC,REGION=4M,PARM=' COPY '
//STEPLIB    DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEIN     DD    DSN=BACKUP.OPEN1,DISP=OLD,VOL=(,RETAIN)
//TAPEOUT    DD    DSN=BACKUP.OPEN1.COPY2,DISP=(,CATLG),
//            UNIT=3490,VOL=(,RETAIN),LABEL=1
//COPY2      EXEC  PGM=FDRSOSTC,REGION=4M,PARM=' COPY '
//STEPLIB    DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEIN     DD    DSN=BACKUP.OPEN2,DISP=OLD,VOL=(,RETAIN)
//TAPEOUT    DD    DSN=BACKUP.OPEN2.COPY2,DISP=(,CATLG),
//            UNIT=3490,VOL=(,RETAIN,REF=*.COPY1.TAPEOUT),LABEL=2
//COPY3      EXEC  PGM=FDRSOSTC,REGION=4M,PARM=' COPY '
//STEPLIB    DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEIN     DD    DSN=BACKUP.OPEN3,DISP=OLD,VOL=(,RETAIN)
//TAPEOUT    DD    DSN=BACKUP.OPEN3.COPY2,DISP=(,CATLG),
//            UNIT=3490,VOL=(,RETAIN,REF=*.COPY2.TAPEOUT),LABEL=3
```

220.05 FDRSOS OPEN SYSTEM UTILITIES

Several utilities are provided which run on the Open System to provide functions such as mounting and dismounting volumes. Documentation is found in the Appendix of this manual.

220.10 FDRSOS SPECIAL CONSIDERATIONS

The following sections detail some special considerations and procedures for the use of FDRSOS, including:

- Coordination of FDRSOS backups and restores with operations on the Open System.
- Coordination of FDRSOS backups with other backup products
- Use of FDRSOS backups with Innovation's FDR/UPSTREAM LAN-to-MVS backup product
- Recovering individual files from FDRSOS backups
- Diagnostic techniques, especially for verifying and correcting connectivity between the S/390 system and the EMC Symmetrix Open System devices.

220.11 FDRSOS COORDINATION

Since there is no communication between FDRSOS and the Open System (UNIX, Windows NT, etc.) actually using the data, some manual coordination may be necessary to ensure that you have valid backups and can successfully do restores.

**QUIESCING
OPEN
SYSTEM
DATA**

In general, you must ensure that the data on the Open System volumes has been completely written to disk (called "hardening") and is not being updated while the FDRSOS backups are running ("quiesced").

Data on Open System volumes may be in many formats, including various file system formats and database formats. Remember that an FDRSOS backup is a physical backup, reading the volume from the lowest addressed data block to the highest; it is not aware of file structures or database formats. Since the data belonging to a given file, database or directory may actually be written in several nonadjacent areas of the disk, it is possible that FDRSOS will backup one part of a file and backup a second part of it many minutes later. If the data is updated between those two points, a restore from that backup may create an unusable version of the file. Even worse, if directory information or database indexes are updated during the backup, a great deal of data may be unusable or inaccurate when restored.

Also, some systems may buffer updated data in memory or cache of some sort, not writing it to disk immediately. In this case a backup may not get the most recent version of the data.

In most cases, you must take whatever steps are required on the Open System to see that all buffered data is flushed out and written to disk (hardened), and that no updates are allowed to take place during the backup. The necessary steps will vary depending on the type of system or database involved.

Some database systems are architected to allow updates to take place during backups, using an update journal file to keep track of updates. After a restore, you specify the time that the backup began and the database system will reapply all updates from the journal, even if some of them are already in place from the backup. You must be sure that your database software supports this function (update from journals after a physical volume restore) before you allow updates to occur during the FDRSOS backup; be very careful, not all journaled databases will support this type of restore. Consult with your database vendor and Innovation if you are not sure.

CONTINUED . . .

220.11 CONTINUED . . .

**BACKUP
COORDINA-
TION**

You may need to coordinate your FDRSOS backups to ensure that you get a consistent backup image of the Open System data.

For some types of Open System data, each EMC logical disk (which looks like a SCSI physical disk) is an independent volume, including its own directory and not containing any data which must be coordinated with data on any other disk volume. Such volumes can be backed up and restored independently.

But other types of Open System data is spread across multiple Open Systems volumes, which requires that all Open System volumes involved must be backed up at about the same point-in-time, and all must be restored if any are restored.

An example of this is IBM AIX/6000 volume groups. A volume group is a collection of physical SCSI disks (EMC logical disks in this case) on which logical volumes are defined. These logical volumes (which are treated like physical volumes by the end-user) may be spread across any or all of the physical volumes in the volume group, as a collection of smaller areas called partitions. It is necessary to backup and restore all of the volumes in a given volume group together; otherwise the data may be totally unusable when restored.

Another example is Windows NT "volume sets", where a Windows "drive letter" is actually defined across several physical SCSI disks (or EMC logical disks).

**RESTORE
COORDINA-
TION**

Since FDRSOS will restore all data blocks on the Open System volume, the volumes being restored must usually be dismounted (disconnected) from the Open System CPU. The procedures for doing this will vary depending on the type of system, but it is necessary that you do this or the system may not properly recognize that FDRSOS has restored the volume. For Novell Netware and Windows NT, see the appendix of this manual.

In extreme cases, it may be necessary to shut down the Open System during the restores and restart it when they are complete.

Innovation intends to provide guidance on the restore procedures for each type of system supported.

CONTINUED . . .

220.11 CONTINUED . . .

**HARDWARE
RESERVES**

RESERVE is a disk hardware instruction which is used by one system to prevent other systems from accessing a given disk volume, even though they have a channel connection to the disk device. The lockout continues until the RESERVing system explicitly RELEASEs the device.

The EMC Symmetrix supports two sorts of RESERVEs for Open System disks:

- a S/390 RESERVE
- a SCSI RESERVE

Either kind of RESERVE prevents any other system, whether S/390 or SCSI, from accessing the disk. S/390 I/Os to a RESERVED disk will simply wait, but the issuing job; may eventually time out and abend. SCSI I/Os to a RESERVED disk may cause waits or may cause I/O errors.

To the best of Innovation's knowledge, only IBM's AIX/6000 (UNIX for the RS/6000) issues a SCSI RESERVE to mounted disks. Since the AIX RESERVE would prevent FDRSOS from backing up the volumes, EMC has provided a special I/O technique which will allow FDRSOS to access the Open System volumes even though AIX (or any other system) has a SCSI RESERVE on the volumes.

FDRSOS has a option to issue a S/390 RESERVE on Open System volumes during backups, restores, and prints (the RESERVE= operand). Since this RESERVE will cause SCSI operations on the volumes (even reads) to wait or possibly fail, Innovation does not recommend its use during backups unless you have no other way of inhibiting updates to the volumes during backups. For restores, the default is RESERVE=YES since you generally want to prevent access to the volume during a restore.

Note that if RESERVE=YES is used and the Open System volume is currently RESERVED to a SCSI channel, FDRSOS will fail the operation with an error message; it will not attempt to wait. If an AIX/6000 system has a volume RESERVED, you cannot use the FDRSOS RESERVE unless you unmount the AIX volume group from AIX (the 'varyoffvg' command).

220.12 FDRSOS AND OTHER BACKUPS

FDRSOS backups are primarily for hi-speed full-volume backup and recovery. Depending on the type of data on the Open System volumes being processed by FDRSOS, you may need to use other backup products to provide all of the recovery that you need.

Innovation's FDR/UPSTREAM, a hi-speed LAN-to-MVS backup/recovery system, can provide the necessary additional recovery in many cases. FDR/UPSTREAM is being enhanced to record and coordinate FDRSOS backups with normal FDR/UPSTREAM backups to provide a complete automated backup solution. See [Section 220.13](#) for details on FDRSOS usage with FDR/UPSTREAM.

DATABASES

If the data on the Open System volumes is large database files, you will need to review the recovery options that are provided for the database system you are using.

Many database systems provide logging of all updates, creating a separate log file in which every change to the database is recorded. In this case, recovery of the database is relatively simple: you must restore the FDRSOS backup of every volume involved in the database, then instruct the database system to reapply all updates that were recorded after the FDRSOS backup was taken. However, you will need to provide for separate backups of the log files, especially if you intend to use them at a disaster recovery site. One way to do so is to put the log files on a separate, small logical volume within the EMC Symmetrix system, and backup that log disk with FDRSOS on frequent intervals, while backing up the databases themselves once a day or even less.

Some database backup products allow you to backup only the changed records within a database. This can also be used with FDRSOS backups if the database recovery product allows you to restore only the records that were changed after a given point. You can run FDRSOS backups on some schedule (perhaps weekly) and do the database backup of changed records more frequently (perhaps daily). If recovery is required, restore the most recent FDRSOS backup and then restore the database records that changed after the point of the FDRSOS backup. Note that this is practical only if the update activity in the database is relatively low; a database where a large percentage of the data is changed daily should be backed up daily in its entirety by FDRSOS.

FILE SYSTEMS

Open System volumes containing many smaller files may also need additional backups with a product that can backup only the files that change each day (such as FDR/UPSTREAM, [see Section 220.13](#)).

To recover such a volume you would first do a full-volume restore from the most recent FDRSOS backup, then instruct the other backup product to restore any files that changed after the time of the FDRSOS backup.

Warning: because FDRSOS restores the entire physical volume, all files that existed on that volume at the time of the backup will be restored. If some of those files were deleted from the volume after the FDRSOS backup, they will reappear on the restored disk. In the worse case, this may not leave enough room on the volume for the restore of updated files from the other backup product.

Such a scheme also allows for the recovery of individual files from the backups. Although recovery of individual files from FDRSOS backups may be possible ([see Section 220.14](#)), it is much less convenient than the service provided by a file backup product.

220.13 FDRSOS AND FDR/UPSTREAM**WHAT IS
FDR/
UPSTREAM?**

FDR/UPSTREAM is Innovation Data Processing's LAN-to-MVS file backup and recovery product. It uses SNA/APPC and/or TCP/IP communications to backup files from servers and workstations to disk or tape backup data sets on MVS.

FDR/UPSTREAM supports many of the systems which may be using Open System volumes on EMC Symmetrix, including Windows NT, Novell Netware, OS/2, and AIX/6000. Other versions of UNIX will be supported in the future.

Please consult the FDR/UPSTREAM documentation for complete details on the procedures described below.

**FDRSOS USE
WITH FDR/
UPSTREAM**

FDR/UPSTREAM V2.5 and above can be used with FDRSOS backups for Open System disks which contain file systems:

- The Open System volumes should be backed up with a FDR/UPSTREAM first-time full MERGE backup, one time. This will backup all files on the volumes to provide a baseline backup for future MERGE backups.
- FDR/UPSTREAM incremental backups to disk should be done on a regular basis, probably daily.
- On a regular basis, perhaps weekly, USTMIGRT should be run with the FORWARD option. This will move any incrementals on disk to tape, plus it will copy all previous incrementals to the same output tape.

Since FDR/UPSTREAM will always have the current backup of individual files on the volume, individual files can be restored at any time.

For full-volume recovery, you must first restore the most recent FDRSOS full-volume backup. FDR/UPSTREAM V2.5.4 and above contains a "Restore back to FDRSOS Full" option. When enabled by an option in FDR/UPSTREAM on the Open System, it records a file containing the date/time of the last FDR/UPSTREAM backup on the Symmetrix volume. This timestamp file is restored by FDRSOS (along with the rest of the data on the volume). When the "Restore back to FDRSOS Full" option is selected, the FDR/UPSTREAM restore recognizes that timestamp and restores only backups created after that time automatically.

As a convenience, the FDR/UPSTREAM-MVS ISPF panels support an option to quickly generate FDRSOS batch jobstreams. One or may volumes may be selected, and it provides flexible options for stacking backups on tape.

FDR/UPSTREAM V2.5.5 has been further enhanced to support FDRSOS:

- FDR/UPSTREAM can now record the backups taken under FDRSOS. The backups must still be done by batch jobs independently of FDR/UPSTREAM, but after the backups are complete, a FDR/UPSTREAM utility (USTREGEN) can be run to read each of the backups and create an entry in the FDR/UPSTREAM repository documenting the date, time, and disk volume serial of each backup.
- If a restore of an Open System volume is required, the best way to do it is by a FDRSOS full-volume restore, using a batch job as described in this manual. However, it may not be possible to do a FDRSOS restore, especially at a disaster site where the disks may not be connected to the MVS host, or the restore disk may not even be EMC Symmetrix disks. In these cases, FDR/UPSTREAM can do a "raw" restore from the FDRSOS backup, transmitting the data over the network where it is rewritten by FDR/UPSTREAM on the workstation. This provides the equivalent of a FDRSOS restore, but at much lower speed.

CONTINUED . . .

220.13 CONTINUED . . .

**REMOTE
INITIATED
FUNCTIONS**

The USTBATCH function of FDR/UPSTREAM allows a batch job or TSO user on the MVS mainframe to initiate functions on the workstation, server or Open System including:

- FDR/UPSTREAM backups and restores
- programs or batch files

The latter allows you to run a MVS job which can execute the necessary procedures on the server or Open System to close databases, mount volumes, or whatever functions are required to prepare for FDRSOS or FDR/UPSTREAM operations. This can include the utility programs described in [Section 220.05](#).

220.14 FILE RESTORES FROM FDRSOS BACKUPS

Although FDRSOS is designed for full-volume backup and recovery of Open System volumes, it is possible to recover individual files from FDRSOS backups in many cases. This procedure **must be tested** in your environment to insure that it works.

ALTERNATE VOLUME RESTORE

To perform this file recovery, you must reserve "spare" logical Open System volumes in the EMC Symmetrix system on which you can restore the FDRSOS backups of the "live" volumes from which you wish to recover. In other words, you must be able to restore the backup to a logical volume which is not currently in use but which can be accessed by the Open System to which the data belongs.

Obviously the spare volumes must be equal in size to the volumes whose backups you are restoring. If the data involved is spread across several Open System volumes (e.g., a AIX/6000 Volume Group), then you must have spare volumes of the proper number and size to restore all of the Open System volumes involved.

Most Open Systems record a volume identification (sometimes called a PVID or signature). In various locations on an Open System volume; this is similar to a MVS volume serial, but it is usually assigned by the operating system, not by the user. The volume identification may be used by the Open System software to identify the volume when it is mounted (or during IPL). In most cases, the Open System will not allow 2 volumes with the same identification to be mounted. Note that some volume formats, such as "raw volumes" and some database formats, do not have volume identifications.

Since FDRSOS normally backs up and restores all data on the volume, including the identification values, this would mean that a normal restore to an alternate volume would make the alternate volume unusable since the original volume with the same identifier is still mounted. To address this, FDRSOS has special support for alternate volumes:

- first, you must format the target volumes (the alternate volumes) with the appropriate Open System utility. If the system supports various disk formats (different types of file systems), it must be formatted as the same type as the volumes whose backup you are about to restore. The volumes should be the same size as the original volumes. This will assign unique identifiers to the volumes.
- second, you must specify the VOLRESET=NO operand on the RESTORE statement for the alternate volume restore (see [Section 210.03](#)). With VOLRESET=NO, FDRSOS will attempt to identify the format of the output volumes, locate the volume identification fields on the volumes, and read them. Then, as the backups are being restored, the identifier value preserved will be inserted into the identifier fields of the appropriate data sectors. In other words, the target volumes will still have the same unique identifiers after the restore. If the IDs are not valid on the disk, FDRSOS will take the ID from the backup and make it unique by adding an increment. The Open System will see these volumes as having identifiers distinct from the original volumes whose backup you just restored, and will be able to access them. VOLRESET=NO will also modify some other identification fields, such as logical volume IDs and UNIX mount points, to make them unique.

220.14 CONTINUED. . .

Currently, VOLRESET=NO only supports volumes with certain formats:

- volumes used by AIX/6000 (UNIX for IBM RS/6000 systems). VOLRESET=NO will change the PVID on the physical volumes and the VGID (volume group ID) associated with each logical volume. It will also attempt to identify the "mount point" name associated with each logical volume (if any) and modify it by adding " SOS" to the end of the name, allowing it to be mounted even though the original volumes are still mounted.
- volumes used by HP/UX are processed similarly to AIX volumes. If the volumes contain a Veritas file system, there are no mount points; for a standard HP file system, mount points are modified as shown above.
- volumes used by Sun Solaris systems. The volume ID and label are preserved from the output disk.
- volumes used by Windows NT.
- volumes used by Novell Netware

**MOUNT THE
RESTORED
VOLUMES**

Once the Open System volumes are restored to the spare volumes, you must make those alternate volumes accessible to the Open System. The details of this will vary depending on the type of system. On some, they may simply appear as extra "drive letters" (e.g., the K: drive). On some, special procedures are required to "mount" the volumes. For systems where no native commands to do this function are provided, the Innovation utilities documented in the Appendix of this manual provide the proper functions for most Open Systems.

**COPY THE
FILES**

Once the restored volumes are mounted, the files on them can be accessed. The required files should be copied to the proper locations on the normal volumes, using normal copy utilities. If necessary, the original files should be overlaid.

**DISMOUNT
THE
VOLUMES**

Once the files have been recovered, the spare volumes should be "dismounted", if such a procedure exists on the system, to insure that files on the restored volume are not accidentally accessed in place of the live files. For systems where no native commands to do this function are provided, the Innovation utilities documented in the Appendix of this manual provide the proper functions for most Open Systems.

TESTING

WARNING: Because of the many variables involved, Innovation cannot be sure that this procedure will work in your environment. You should test this process on your system before you rely on it for file restores.

220.15 FDRSOS DIAGNOSTIC TECHNIQUES

This section primarily concerns diagnosis of problems which sometimes occur during the initial installation and configuration of an EMC Symmetrix for use with FDRSOS. It discusses the different types of problems which can occur, documents some diagnostic tools, and suggests solutions to common problems.

It assumes that you (and your EMC representative) believe that you have taken all the proper steps, outlined in [Section 200.03](#), to configure the Symmetrix, attach it to the S/390 system, and update the S/390 I/O configuration.

**MVS
CONSOLE
COMMANDS**

The first tool for checking the configuration and connection is the MVS DEVSERV command (abbreviated DS) on the console. It will indicate whether MVS can successfully connect to the Symmetrix control unit and to the Open System volumes in particular, without involving FDRSOS.

You can verify the connectivity from MVS to the Symmetrix with the DEVSERV PATHS command (abbreviated DS P). If your Open System volumes were at addresses 1E0-1EF, you could issue the command:

```
DS P,1E0,16
```

which would produce a display similar to:

```

      IEE459I 16.49.32 DEVSERV PATHS 321
UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
01E0,3390   ,F,000,E#01E0,20=+ 21=<
01E1,3390   ,F,000,E#01E1,20=+ 21=<
01E2,3390   ,F,000,E#01E2,20=& 21=<
.....
***** SYMBOL DEFINITIONS
F = OFFLINE
< = PHYSICALLY UNAVAILABLE
+ = PATH AVAILABLE
& = RESERVED TO ANOTHER PATH/GROUP
* = LOGICALLY OFF, PHYSICALLY ON
- = LOGICALLY OFF, PHYSICALLY OFF

```

It displays:

- the device address
 - the device type (make sure it is 3380 or 3390)
 - device status (make sure it is F for offline)
 - the volume serial (which will be blank unless you have previously used the volume with FDRSOS)
 - the channel paths (CHPIDs) defined for this device (up to 4) and the status of each.
- A good connection is indicated by a status of:

+ path available

& path available but the device is currently reserved to another system, such as a AIX/6000 system

220.15 CONTINUED. . .

**MVS
CONSOLE
COMMANDS
(continued)**

Review the display to be sure that every path to every device has a good connection (except for paths which have not yet been physically connected, of course). Connection problems might be indicated by:

- * path available but marked offline for this device. This might be due to a VARY PATH console command; VARY PATH is not recommended for Open System volumes. This might also occur if connection errors occurred at IPL time or when the device was dynamically added to the configuration. The FDRSOS command VARYPATH may be able to reactivate these paths
(See Section 210.07)
- path physically offline to the hardware. You may be able to vary it online with the MVS console command: CF CHP(xx),ONLINE. Functions on the hardware console may also be used to re-enable the CHPID.
- < path online but device could not be contacted, control unit not responding. This is the most common connection error status. See "Connection Errors" below for hints on finding the problem.

Other error status codes are possible but less common.

Another useful console command is the DISPLAY M command (abbreviated D M). It will display the connection status of individual devices or all devices on a path, but unlike DEVSERV, does not actually attempt to verify the connection.

To display individual devices, issue:

```
D M=DEV(1E0)    or    D M=DEV(1E0-1EF)
```

specifying the device address or address range. To display all devices on a channel path in a compact format, issue:

```
D M=CHP(20)
```

specifying the CHPID of the channel. The information displayed is similar to that displayed by the DEVSERV command.

220.15 CONTINUED. . .

**FDRSOS
DIAGNOS-
TICS**

Most of the time, when a device address or address range is specified on a FDRSOS MOUNT statement, e.g.,:

```
MOUNT UNIT=01E4
MOUNT UNIT=01E*
```

the FDRSOS UCB scan routine will ignore any UCBs which are not candidates for Open System volumes. The tests it applies are:

- is the device a disk?
- is the device offline?
- is the device enabled for I/O?
- does the device have at least one enabled path (CHPID)?

This is done in case you specify addresses or ranges which are not Open System volumes on EMC Symmetrix subsystems, so that FDRSOS will not generate useless I/Os and error messages. If it passes the above tests, FDRSOS will then attempt to do I/O to the device, using CCW chains that will work on EMC Open System devices but which will fail on all other devices (including CKD disks from EMC and all other vendors). In most cases, if this I/O fails then FDRSOS will bypass the device, assuming that it is not an Open System volume.

If some configuration error prevents the test I/O from completing successfully on a valid Open System volume, FDRSOS may not report the error, making it difficult to determine what happened. This is done so that you specify a range of device addresses FDRSOS does not generate a large number of error messages for devices which are not Open System volumes. If a given MOUNT statement references multiple devices (e.g., MOUNT UNIT=01E*), FDRSOS will only report results on devices that have been identified as Open System volumes. However, if all devices specified by one MOUNT statement fail to pass the tests, you will get a FDR316 message indicating that no devices matched your MOUNT.

To make it easier to diagnose such problems, there is one exception to the above. When processing a LABEL statement, which requires a MOUNT statement specifying a single unit address, FDRSOS bypasses the above tests and goes straight to the test I/O. If the test I/O fails, then you will get a message and a I/O trace showing the results, similar to this example:

```
FDR303 CARD IMAGE -- LABEL TYPE=SOS
FDR303 CARD IMAGE -- MOUNT UNIT=014F
FDR212* FDRSOS ERROR UNIT=014F REASON=4 - NOT A RESPONDING SOS UNIT
FDR149 I/O TRACE SECT=X'00000000' 3.504 MSEC - 109,367.476 - DD=D#FDRSOS - UNIT=014F
FDR149 CCW/DATA ---- IOB C6008000 41009290 00008BD8 0E000000 00008BD0 5000921C 00000000 00000000
-- SENSE 80000000 CF000004 00000032 95000004 020085BD 03010F04 00004CE3 00000000
008BD0 AF008E08 6000000C E4008EA8 60000007 64008EB0 20000040
1F000000 00000000 00000000 00000000 00000000 00000000
```

220.15 CONTINUED. . .

**FDRSOS
DIAGNOS-
TICS
(continued)**

In the "I/O trace", there are several fields of interest:

- the IOB documents the termination status of the test I/O. The first byte of the second word is the terminating status indicator; it is X'41' in this example, indicating an I/O error. Another possible value is X'6D' indicating no paths were available (CC=3 on the Start Subchannel).
- The third and fourth words of the IOB are the CSW (Channel Status Word) documenting the I/O termination. The first 2 bytes of the fourth word are the device and channel hardware status of the I/O (X'0E00' in our example).
- If the device status include the X'02' flag (Unit Check), a device I/O error occurred, and the SENSE line will be printed showing detailed status information returned by the device. In this example, the sense starts with X'80' (Command Reject) indicating that the control unit did not recognize our test CCW chain, which may mean that it is not properly configured as an Open System device.
- The lines below the IOB and SENSE are the actual CCW chain executed.

Your EMC representative may be able to help you interpret the I/O status indicators. However, you may always call Innovation Technical Support for assistance if you are unable to determine the cause.

220.15 CONTINUED...

**FDRSOS UCB
DISPLAY**

FDRSOS has another diagnostic tool built in, a UCB display. Since FDRSOS does a number of tests against the UCB (Unit Control Block) to determine if it is a valid Open System volume, some problems can be diagnosed by examining the fields in the UCB. In fact, if you are calling Innovation Technical Support because of connection problems, we would prefer that you get this UCB display and have it available when you call.

To get the UCB display, simply add the parameter PRINT=UCB onto your first FDRSOS control statement. This will cause FDRSOS to format the UCB of every device address you specify. The UCB is formatted in the joblog and system messages of the FDRSOS job. In ESA 5.1 and above (including OS/390) it will be similar to (some sections abbreviated to conserve space):

```
UCB PREFIX AT 01E49BD0
-0008 LOCK..... 00000000 IOQ..... 00000000
UCB AT 01E49BD8
+0000 JBNR..... 00 FL5..... 88 ID..... FF
+0003 STAT..... 04 CHAN..... 01FF FL1..... 00
+0007 FLB..... 00 NXUCB..... 00000000 WGT..... 04
+000D NAME..... 1FF TBYT1..... 30 TBYT2..... 30
+0012 DVCLS..... 20 UNTYP..... 0F FLC..... 00
+0015 ETP..... E49BB1
+0018 VTOC..... 00000000 VOLI..... STAB..... 00
+0023 DMCT..... 00 SQC..... 00 FL4..... 00
+0026 USER..... 0000
UCB COMMON EXTENSION AT 01E49BB0
UCB EXTENDED PREFIX AT 01E50080
+0000 RSTEM..... 00 MIHKY..... 04 MIHTI..... 00
+0003 HOTIO..... 00 IOQF..... 00000000 IOQL..... 00000000
+000C SIDA..... 0001 SCHNO..... 0090 PMCW1..... 2898
+0012 MBI..... 00A0 LPM..... C0 RSV..... 00
+0016 LPUM..... 10 PIM..... C0 CHPID..... 2021FFFF
+001C ..... FFFFFFFF LEVEL..... 01 IOSF1..... 40
+0022 IOTKY..... 00 MIHFG..... 00 LVMSK..... 00000001
DCE AT 01E49B80
CMB AT 053FE400
```

On systems before ESA 5.1, the Extended Prefix does not exist, so those fields are in the prefix, as shown below:

```
UCB PREFIX AT 00FCECE0
-0030 RSTEM..... 00 RSV..... 00 MIHTI..... 00
-002D HOTIO..... 40 IOQF..... 00000000 IOQL..... 00000000
-0024 SIDA..... 0001 SCHNO..... 0010 PMCW1..... 2898
-001E MBI..... 0018 LPM..... F0 RSV..... 00
-001A LPUM..... 10 PIM..... F0 CHPID..... 01020607
-0014 ..... 00000000 LEVEL..... 01 IOSF1..... 00
-000E IOTKY..... 00 MIHFG..... 00 LVMSK..... 00000001
-0008 LOCK..... 00000000 IOQ..... 00FC4880
UCB AT 00FCED10
+0000 JBNR..... 00 FL5..... 8A ID..... FF
+0003 STAT..... 84 CHAN..... 014F FL1..... 40
+0007 FLB..... 00 NXUCB..... 00FCEDB0 WGT..... 04
+000D NAME..... 14F TBYT1..... 30 TBYT2..... 30
+0012 DVCLS..... 20 UNTYP..... 0E FLC..... 04
```

220.15 CONTINUED. . .**FDRSOS UCB
DISPLAY
(continued)**

Some things that FDRSOS checks:

- in field STAT, X'80' indicates it is online; this must not be on
- field DVCLS must be X'20' (disk device class)
- in field FLB, the X'40' or X'20' bits indicate no path to device; they must be off
- field LPM (logical path mask) must be non-zero otherwise there are no enabled paths (CHPIDs) to the device.

You can verify these fields in the UCB display. If the UCB fails one of the first 2 checks, check your I/O configuration (HCD or MVSCP). If it fails the last 2 tests, see "Connection Errors" below.

Normally the LPM and PIM (path installed mask) are the same. The field CHPID shows the CHPIDs assigned to this device.

220.15 CONTINUED. . .

**FDRDEBUG:
ONE MORE
TOOL**

FDRSOS includes a utility debugging program called FDRDEBUG; it is also distributed with FDR. FDRDEBUG has a variety of functions, some of which are useful with FDRSOS.

FDRDEBUG can be called from TSO or executed as a batch program. From TSO enter:

```
CALL 'fdrsos.library(FDRDEBUG)' 'FDRDEBUG,function'
```

In batch, execute:

```
//DEBUG EXEC PGM=FDRDEBUG,PARM='FDRDEBUG,function'
//STEPLIB DD DISP=SHR,DSN=fdrsos.library
```

no SYSPRINT is required; all displays are done to the job log of the job.

The useful functions include:

LLA	display linklist libraries
APF	display APF authorized libraries
modulename	locate the named module
?hexaddress	locate module at that address
*hexaddr,len	display storage at that address
-device	display UCB for the named device (e.g., -1E4 or -374E)
_device	monitor CMD for the named device

For example:

```
CALL 'fdrsos.library(FDRDEBUG)' 'FDRDEBUG,APF'
//DEBUG EXEC PGM=FDRDEBUG,PARM='FDRDEBUG,-1E8'
//STEPLIB DD DISP=SHR,DSN=fdrsos.library
```

The last function (_device) is useful to monitor the performance of an Open System volume once it is operating normally. Every 5 seconds it displays fields from the CMB (Channel Measurement Block) allowing you to see how performance is changing. It terminates after 5 minutes or when you do ATTENTION under TSO. The display will be similar to:

UNIT	VOLSER	SECS	CONN	PEND	DISC	CUQ	-	TOTC	TOTP	TOTD	TOTQ
1EFC	001EFC	0005	46%	1%	52%	0%		40%	0%	60%	0%
1EFC	001EFC	0005	47%	1%	53%	0%		40%	0%	60%	0%
1EFC	001EFC	0005	47%	0%	53%	0%		40%	0%	60%	0%
1EFC	001EFC	0005	45%	0%	52%	0%		40%	0%	60%	0%

The Connect, Pending, Disconnect and CU Queue values are the percentage of active I/O time spent in each state in the preceding 5 second interval; the equivalent TOT fields are values since the last IPL. In this example of a real FDRSOS backup, disconnect is mostly time spent by the Symmetrix fetching disk data into the cache, and connect is time spent transferring data to the channel. Larger values for pending (waiting for I/O initiation) or CU queue (waiting for free control unit) might indicate that the channels to the Symmetrix are overloaded. Larger values for disconnect may indicate contention problems within the Symmetrix.

220.15 CONTINUED. . .

**CONNECTION
ERRORS**

These hints may help you resolve your connection problems. If they do not, please contact your EMC representative and Innovation Technical Support for assistance.

If all devices on the Symmetrix subsystem are failing (including any S/390 CKD disks in the same subsystem):

1. Make sure that the physical channel connections have been made properly (ESCON or parallel channels properly connected on each end).
2. Check that the proper channel paths (CHPIDs) have been specified in your I/O configuration.
3. If your Symmetrix is connected by an ESCON director, be sure that the director configuration is correct.
4. Ask your EMC representative to verify that the Symmetrix configuration is correct.

If CKD disks in the same Symmetrix subsystem are working but all of the FBA Open System volumes are failing:

5. Check your I/O configuration (HCD or MVSCP) to be sure that they are properly defined as 3380 or 3390 disks and are marked offline.
6. Ask your EMC representative to be sure that the FBA disks in the Symmetrix have been assigned both a S/390 address and a SCSI address.

If the Open System volumes are assigned only a S/390 address, FDRSOS support in the Symmetrix microcode will not be enabled and they will respond only to standard IBM FBA CCWs, not the special CCWs used by FDRSOS. This will result in a Command Reject as shown in the sample I/O trace earlier in this section. If they are assigned only a SCSI address, they will not respond to the S/390 at all. They must be assigned both a SCSI and S/390 address to be eligible for FDRSOS processing.

If the DEVSERV PATH command shown earlier in this section shows a status of < for a path, this means that the control unit (the Symmetrix channel adapter) responds to the device address, but indicates that no such device exists. This is similar to a short string of IBM disks where some device addresses do not exist. This is probably due to some error in the Symmetrix configuration.

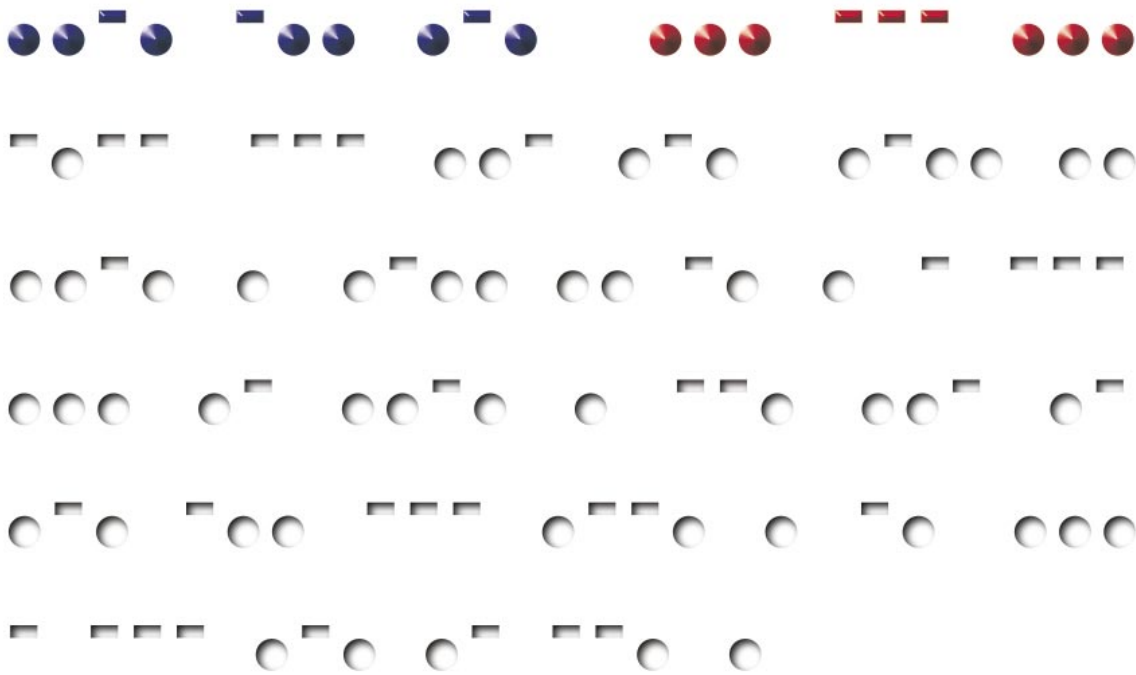
If I/O errors are occurring on the Open System volumes for no obvious reason, you can try the FDRSOS VARYPATH command ([See Section 210.07](#)); it resets certain flags which may prevent I/O from completing successfully, and reenables all defined paths to the device.

As always, if you and EMC cannot determine the problem, please call Innovation Technical Support.

I/O ERRORS

Because of the high reliability of the EMC Symmetrix subsystem, and additional recovery provided by the mirroring and RAID-S options of the subsystem, unrecoverable I/O errors will be rare. If they do occur, you will receive an I/O trace printout similar to that shown earlier in this section. The IOB, sense, and CCW printouts will identify the error that occurred. You can discuss these errors with your EMC representative, or call Innovation Technical Support.

THIS PAGE WAS INTENTIONALLY LEFT BLANK



Installation and User's Guide



THIS PAGE WAS INTENTIONALLY LEFT BLANK

230.01 FDRSOS INSTALLATION

Section 230 contains the simple instructions for installing the FDRSOS product on your system.

**DISTRIBUTION
TAPE
FORMAT**

FDRSOS is distributed on a standard label (SL) magnetic tape volume. The volume serial and status of the distribution tape will be clearly marked on the external tape label. The volume serial of the tape will be:

FDR53P - a production (licensed) version of FDRSOS.

FDR53T - a trial (evaluation) version of FDRSOS. The trial programs will expire (stop functioning) on the expiration date indicated on the external label.

The files on the tape are:

File 2: DSN=LOAD

IEBCOPY-unloaded library of programs.

File 4: DSN=ICL

IEBCOPY-unloaded library of installation jobstreams and supplementary documentation.

File 10: DSN=JCL

IEBCOPY-unloaded library of all example JCL from this user manual.

Since the FDRSOS distribution tape shares a common format with other Innovation distribution tapes, many files on the tape are not used at this time. Any files not listed above will have a data set name of DUMMY and will be empty.

**INSTALLATION
INSTRUCTIONS**

The following sections document installation of the FDRSOS libraries.

WARNING: FDRSOS customers who are also customers of Innovation's FDR products, including FDR, FDRDSF, COMPAKTOR, FDREPORT, FDRREORG, and/or FDR/ABR, must install FDRSOS in a library separate from that containing any of the other FDR products. If the FDR products are in a system linklist library, you cannot put FDRSOS in the linklist as well; you must use a STEPLIB in all FDRSOS jobstreams.

230.02 INSTALLING THE INSTALLATION CONTROL LIBRARY

File 4 of the installation tape contains IEBCOPY input that will create an Installation Control Library. This library supplies the user with JCL streams and supplementary documentation. The member named 'INDEX' contains a brief functional description of each member within the Installation Control Library.

DISK SPACE REQUIRE- MENTS

The Installation Control Library must be loaded to a partitioned data set on disk. You may load it to an existing data set (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the Installation Control Library:

PDS DIR					
DATA SET	RECFM	LRECL	BLKSIZE	BLOCKS	BLOCKS
IDP.ICLFDR53	FB	80	3120	200	10

This data set is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this data set, you should adjust the number of blocks accordingly.

INSTALLA- TION INSTRUC- TIONS

Execute the following JCL to load the Installation Control library. The following changes must be made to reflect your environment:

- 'DSN=IDP.ICLFDR53' on the SYSUT2 DD statement should be changed to the name you wish to use for the Installation Control Library.
- 'VOL=SER=vvvvvv' on the SYSUT2 DD statement must specify a disk volume where the Installation Control Library will be allocated, or may be omitted if the data set will be SMS-managed.
- 'UNIT=TAPE' on the SYSUT1 statement must specify a tape drive capable of reading the installation tape that you have received.
- 'VOL=SER=FDR53T' on the SYSUT1 statement must be changed to 'VOL=SER=FDR53P' if you are loading from a production installation tape.

JOB CONTROL FOR LOADING LIBRARY

```
//ICLLOAD      EXEC  PGM=IEBCOPY,REGION=1024K
//SYSUT2       DD   DSN=IDP.ICLFDR53,          <--USER-CHANGE
//              VOL=SER=vvvvvv,                <--USER-CHANGE
//              UNIT=SYSDA,DISP=(,CATLG),        SEE NOTE 1
//              DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//              SPACE=(3120,(200,20,10),,,ROUND)
//SYSUT1       DD   DSN=ICL,
//              UNIT=TAPE,                      <--USER-CHANGE
//              DISP=OLD,LABEL=(4,EXPDT=98000),
//              VOL=SER=FDR53T  CHANGE T TO P IF PRODUCTION TAPE
//SYSIN        DD   *
               COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))
```

NOTE 1:To load the members to an existing library,

```
//SYSUT2       DD   DISP=OLD,DSN=IDP.ICLFDR53 <--USER-CHANGE
```

CONTINUED . . .

230.02 CONTINUED . . .

MEMBERS The members in the Installation Control Library provide you with jobstreams for completing the installation, as well as supplementary documentation.

230.03 INSTALLING THE PROGRAM LIBRARY

INSTALLATION

The FDRSOS program library must be loaded to a partitioned data set on disk. You may load it to an existing dataset (if it has sufficient space and proper DCB) or allocate and load a new one. The following table shows the allocation parameters for the program library:

DISK SPACE REQUIREMENTS

DATA SET	RECFM	LRECL	BLKSIZE	BLOCKS	PDS DIR BLOCKS
IDP.MODFDR53	U	n/a	6144	240	34

This data set is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this data set, you should adjust the number of blocks accordingly.

JOB CONTROL FOR LOADING LIBRARY

JCL similar to that below is supplied on the Installation Control Library ([see section 230.02](#)) with a member name of 'SOSLOAD'. The JCL in the Installation Control Library reflects the type of product installation tape from which it was loaded.

```
//LOAD      EXEC      PGM=IEBCOPY,REGION=1024K
//SYSPRINT   DD        SYSOUT=*
//SYSUT2     DD        DISP=SHR,DSN=IDP.MODFDR53          <- - CHANGE
//SYSUT1     DD        DSN=LOAD,
//           VOL=(,RETAIN,SER=FDR53T),UNIT=TAPE, <-CHANGE TO FDR53P IF PRODTAPE
//           DISP=OLD,LABEL=(2,EXPDT=98000)
//SYSIN      DD        *
COPYMOD     OUTDD=SYSUT2,INDD=((SYSUT1,R)),MAXBLK=6144
/*
```

*** CAUTION CAUTION CAUTION ***

1. Because FDRSOS shares some common modules with Innovation's FDR Dasd Management System which may be at a different level than your current FDR modules, you must install FDRSOS in an authorized library separate from your current FDR library, if your installation has any of the components of FDR (FDR, COMPAKTOR, ABR, FDRREORG, and/or FDREPORT). The FDRSOS library should be used as a STEPLIB when required and not placed in the system link list.
2. If you are installing a production tape (which has no date protected modules), you must change the installation JCL to reflect this; change the serial from FDR53T to FDR53P. You may need to change UNIT=TAPE to a unit name capable of reading the distribution tape you received (reel or cartridge).
3. The FDRSOS program library must be an APF authorized library. A library is APF authorized if it is accessed via the LINKLIST or its name is in SYS1.PARMLIB member IEAAPFxx. Under MVS/XA, MVS/ESA and OS/390, if the LINKAUTH=APFTAB option is specified in IEASYSxx, then even a library accessed via the LINKLIST has to be listed in IEAAPFxx to be authorized. For testing purposes, it is possible to dynamically authorize a library.

230.04 INSTALLING THE JCL LIBRARY

File 12 of the installation tape contains IEBCOPY input that will create a JCL Library. This library supplies the user with copies of all of the example JCL shown in this manual.

DISK SPACE REQUIRE- MENTS

The JCL Library must be loaded to a partitioned data set on disk. You may load it to an existing data set (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the Installation Control Library:

DATA SET	RECFM	LRECL	BLKSIZE	BLOCKS	PDS DIR BLOCKS
IDP.JCLFDR53	FB	80	3120	200	10

This data set is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this data set, you should adjust the number of blocks accordingly.

INSTALLA- TION IN- STRUCTIONS

Execute the following JCL to load the Installation Control library. The following changes must be made to reflect your environment:

- DSN=IDP.JCLFDR53' on the SYSUT2 DD statement should be changed to the name you wish to use for the JCL Library.
- 'VOL=SER=vvvvvv' on the SYSUT2 DD statement must specify a disk volume where the JCL Library will be allocated, or may be omitted if the data set will be SMS-managed.
- 'UNIT=TAPE' on the SYSUT1 statement must specify a tape drive capable of reading the installation tape that you have received (reel or cartridge).
- 'VOL=SER=FDR53T' on the SYSUT1 statement must be changed to 'VOL=SER=FDR53P' if you are loading from a production installation tape.

JOB CONTROL FOR LOADING LIBRARY

The JCL below is supplied on the Installation Control Library ([see section 230.02](#)) with a member name of 'SOSJCL'. The JCL in the Installation Control Library reflects the type of product installation tape from which it was loaded.

```
//JCLLOAD      EXEC PGM=IEBCOPY,REGION=1024K
//SYSPRINT     DD  SYSOUT=*
//SYSUT2       DD  DSN=IDP.JCLFDR53,                <- -USER-CHANGE
//              VOL=SER=vvvvvv,                     <- -USER-CHANGE
//              UNIT=SYSDA,DISP=(,CATLG),            SEE NOTE 1
//              DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//              SPACE=(3120,(200,20,10),,,ROUND)
//SYSUT1       DD  DSN=JCL,
//              UNIT=TAPE,                            <- -USER-CHANGE
//              DISP=OLD,LABEL=(10,EXPDT=98000),
//              VOL=SER=FDR53T  CHANGE T TO FDR53P IF PRODUCTION TAPE
//SYSIN        DD  *
               COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))
```

NOTE 1: To load the members to an existing library,

```
//SYSUT2       DD  DISP=OLD,DSN=IDP.JCLFDR53        <- -USER-CHANGE
```

230.10 FDRZAPOP GLOBAL OPTION CHANGE FACILITY

Some options which affect FDRSOS may be changed permanently. All such options are kept in a load module called FDROPT, in the FDRSOS program library. Some of these options affect the defaults for various operands on FDRSOS control statements and can be overridden at execution time; others may be changed only in FDROPT and cannot be overridden.

All options may be set with program FDRZAPOP, the Global Option Change facility.

Note that FDRZAPOP and FDROPT are also used with the FDR system. Many of the options that can be modified or printed by FDRZAPOP have no meaning for FDRSOS. Only those options documented in the following sections are valid for FDRSOS.

**BATCH
EXECUTION**

To execute FDRZAPOP as a BATCH job use the following JCL:

**EXEC
STATEMENT**

Must specify the program name of the Global Option Change Facility -- FDRZAPOP.

**STEPLIB or
JOBLIB DD
STATEMENT**

Must specify the load module library in which FDRSOS resides.

**SYSPPRINT DD
STATEMENT**

Specifies the output message data set. This is a required DD statement and usually is a SYSOUT data set.

**SYSLIB DD
STATEMENT**

Must also specify the load module library in which FDRSOS resides. The user must have authority to update this library.

**SYSIN DD
STATEMENT**

Specifies the control statement data set required for all functions. Usually an input stream or DD * data set.

**TSO EXECU-
TION**

FDRZAPOP can be executed under TSO. The program will prompt you for the commands. The 'END' command will save the modified options, if any, and terminate the program. The 'CANCEL' command will discard any modified options and terminate. The allocations required to execute FDRZAPOP in the TSO Foreground are as follows:

```
ALLOC      F(SYSLIB)      DA('fdrsos.library') SHR
ALLOC      F(SYSPPRINT)   DA(*)
ALLOC      F(SYSIN)       DA(*)
CALL 'fdrsos.library(FDRZAPOP)'
```


230.11 FDRZAPOP STATEMENTS AND OPTIONS

**AVAILABLE
FUNCTIONS**

FDRZAPOP accepts the following statements:

HELP – The HELP command will print a menu of the FDRZAPOP options and related documentation. Note that it will include all FDR options, not just FDRSOS options.

The HELP command format is:

HELP or HELP SUB(command) or HELP ALL

PRINT – The current values in the FDRSOS Global Options Table will be printed. Note that the report will contain all FDR options, not just FDRSOS options.

The PRINT command format is:

PRINT

RESET – The FDRSOS Global Options Table will be reset to the original values on the installation tape.

The RESET command format is:

RESET

ZAP – Modify options in the FDRSOS Global Options Table. This command enables the installation to enable or disable specified functions and set control statements defaults and other processing options for FDRSOS. The operands for ZAP are documented below.

The ZAP command format is:

**ZAP [ENABLE=(option1,...,optionn)]
[,DISABLE=(option1,...,optionn)]
[,operand=value,...,operand=value]**

230.11 CONTINUED. . .

**ZAP
OPTIONS**

These are the operands that can be specified for the ZAP command, to set various permanent options for FDRSOS. If specified as a subparameter of the ENABLE operand, the option is enabled; specifying them as a subparameter of DISABLE disables them.

ALLCALL if enabled, FDRSOS does RACF-compatible security checks for dump, restore and erase operations. For each Open System volume, it issues a security check with CLASS=DASDVOL and ENTITY=the Open System volume serial assigned by the FDRSOS LABEL function. READ authority will be required for backups, and ALTER authority for restore and erase. ALLCALL is disabled by default. To change it, enter ZAP ENABLE/DISABLE=ALLCALL

ALLCALL security checking is effective only if your installation has implemented the DASDVOL security class (which can also be used to control access to normal MVS volumes) and defined the Open System volume serials under that class. FDRSOS will permit all access to a volume if the DASDVOL class is not active or if the volume is not controlled under that class. Consult your security system documentation for guidance on defining volume rules that are compatible with this DASDVOL check.

Note that FDRSOS also supports a volume security exit in which you can program your own security checks for volume access. If you need this function, contact Innovation for more information.

BUFNO As a default, during a DUMP FDRSOS will acquire 16 32K data buffers (ZAP BUFNO=MAX). You can specify that a smaller number of buffers be used (ZAP BUFNO=nn) but this will impact FDRSOS performance. Since this can be overridden at execution time by the BUFNO= operand, Innovation does not recommend changing this option.

LINECNT You can override the maximum number of lines to be printed on any report page produced by the FDR system. The default is 58 lines per page. To change this default, specify ZAP LINECNT=nn, where nn may be a number from 10 to 99.

**MESSAGE
ROUTING AND
DESCRIPTOR
CODES**

The routing and descriptor codes used for the operator console messages in the FDR system can be changed. The default value for the routing codes is (2,11), and for the descriptor code is 2. The routing and descriptor codes are documented in the IBM manual Routing and Descriptor Codes and are used by MVS to control which consoles the messages will appear on, and how they will be displayed. To change them, specify ZAP ROUTECODE=(nn,...,nn) and/or ZAP DESCRIPTCODE=(nn,...,nn).

230.20 FDRSOS MESSAGES

This section describes the various printer and console messages which may be output by FDRSOS system and the various ABEND codes with which it may terminate.

**FDRSOS
MESSAGE FORMAT**

The general format of FDRSOS messages and WTORs is as follows:

FDRnnn	Message-text.
or	
FDRnnn**	Message-text.

When ** appears the message is an error message. The step will ABEND, or will pass a return code of 8 or higher unless indicated otherwise in the specific message.

RETURN CODES

FDRSOS always passes a return code (step completion code) at the end of the step, unless it ABENDs. A return code of zero indicates that FDRSOS has performed all functions successfully. A return code of 4 indicates that the testing period for a trial version of the product has expired. Any other return code indicates that errors of some kind have occurred during this execution of FDR. Check the listing for the error messages.

ABEND CODES

FDRSOS ABEND codes range from a U0100 to a U0900. Most user ABENDs are preceded by an FDR message. Note that some FDRSOS ABENDs will result in the entire FDRSOS step abending (for errors such as invalid JCL and control statement errors), while some ABENDs only affect a subtask under FDRSOS (such as a dump or restore subtask). Subtask ABENDs will be documented by a FDR319 message but they will usually not cause the entire step to abend; other subtasks may continue to execute and the step will end with a non-zero return code to indicate that some unusual condition occurred.

230.21 FDRSOS CONSOLE MESSAGES

The following may appear on an MVS operator's console during FDRSOS operations.

FDRW01 CONFIRM REQUEST TO function RESTORE UNIT=uuuuu,VOL=vvvvvvv,JOB=jjjjjjjj

Reason: The job indicated in the message wishes to restore or erase (as indicated by "function") the Open System volume specified. This confirmation can be suppressed by specifying the "CONFMESS=NO" operand on the RESTORE statement or "CONFERASE=NO" operand on the ERASE statement.

Action: Reply YES—Operator approves the operation.
Reply NO—Operator does not approve the operation; the operation is terminated. The subtask will terminate with a U0801 ABEND, but other restore or erase requests in the same FDRSOS step may proceed.

FDRW89 FDR - TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN 'nnn' DAYS - PLEASE CONTACT INNOVATION

Reason: This is a trial version of FDRSOS. The number of days the trial will remain active is displayed. The product will stop operating when the trial expires.

Action: When there are 10 or fewer days before the trial is due to expire this message will become non-deletable. Call INNOVATION for assistance.

230.22 FDRSOS PRINTER MESSAGES

These messages may appear in FDRSOS printed output.

**FDR001 FDR SAFEGUARD OPEN STORAGE -- FDRSOS VER v.r/mmb -- INNOVATION DATA
PROCESSING DATE=yy.ddd PAGE nnn**

Reason: This is the FDRSOS page heading, specifying the version level of FDRSOS. "v.r" indicates the version and release (e.g., 5.3), "mm" is a 2-digit number indicating the maintenance level and "b" will be "P" for a production version, "T" for a trial, and "B" for a beta test version.

FDR002 function SUCCESSFULLY COMPLETED VOL=vvvvvv

Reason: The Open System volume with volume serial vvvvvv was successfully processed by the function indicated.

FDR006 function OPERATION CANCELLED BY OPERATOR

Reason: An operator replied 'NO' to an FDRSOS request for approval to restore a disk (message FDRW01). A U0801 Abend follows.

**FDR007 STARTING|ENDING TIME OF function -- hh.mm.ss --UNIT=address,
IN=inputdd ,OUTPUT=outdd1 outdd2**

Reason: Documents the time that FDRSOS began or ended the function indicated. "address" is the S/390 device address of the Open System volume.
"inputdd" is the DD name of the DD name of the Open System volume for DUMP, PRINT, and ERASE; this may be a DISKx name or a DD name dynamically allocated by FDRSOS. For RESTORE, it is the TAPEx DD name of the backup being restored.
"outdd1" is the output DD name. If a duplicate backup was requested, it is shown by DD name "outdd2".

FDR008 OPEN ERROR OR NO DD STATEMENT DD=ddname - function BYPASSED

Reason:

1. A tape DD statement specified by "ddname" was missing. A TAPEDD=x operand appeared on a MOUNT statement but the corresponding TAPEx DD statement was not found.
2. The DD statement SYSPRINx is missing. If MAXTASKS= was specified, there must be a SYSPRINx DD for every TAPEx DD.
3. An OPEN error occurred for the specified ddname. Check the job log for IBM error messages.

Action: The operation associated with the failing DD statement will be bypassed. Other operations in the same FDRSOS step may execute.

230.22 CONTINUED. . .**FDR012 NOT AN FDR type TAPE DSN=dsname**

Reason: 1. The backup dataset input for a restore is not a FDRSOS-created backup.
2. The JCL specifies the wrong tape or disk volume serial numbers, or specifies the serials in the wrong order for a multi-volume backup dataset.
3. The FDRSOS backup file may have been written over.
"type" will be BACKUP if the tape is not in FDR format at all, and SOS if it is in FDR format but is not a FDRSOS backup (i.e., is a regular FDR backup). The data set name printed is the backup data set name. A MINI DUMP is printed displaying the first tape block read, which may help you understand what is really in this backup.

Action: The restore subtask will abend with U0205. If possible, correct the JCL to point to the proper backup and resubmit.

FDR014 BACKUP HAS MORE | LESS SECTORS THAN RESTORE DEVICE - RESTORED TERMINATED|CONTINUING

Reason: During a RESTORE, FDRSOS determined that the size of the Open System volume that was backed up to the input backup data set was a different size than the target Open System volume. A MINI DUMP is printed displaying the FDRSOS header record.

Action: If the backup contained more data sectors, the restore is terminated. If it contained fewer data sectors, the restore will continue but the restored volume may or may not be usable, depending on the system which uses the data.

FDR020 RACF VOLUME PROTECTION FAILED ON VOL=vvvvvv

Reason: A security check was done in the DASDVOL class for the volume serial indicated. The user was not authorized to READ that volser for DUMP and PRINT, or ALTER the volser for RESTORE.

Action: The restore subtask will abend with U0801.

FDR031 DD=ddname - NUMBER OF CHARACTERS EXCEEDS 5 - STATEMENT IGNORED

Reason: FDRSOS encountered a DD statement starting with "DISK" but the total length of the DD name was more than 5 (more than one character after DISK, e.g., DISKAA).

Action: The DD statement is ignored. Correct the JCL and resubmit the job.

FDR032 NO VALID DISK(X) DD STATEMENTS WERE FOUND

Reason: FDRSOS did not find any Open System volumes to process. The volumes to process must be specified either by DISKx DD statements or MOUNT VOL= statements. If MOUNT statements were used it may be that the volume serials were misspelled, or the volumes have not been made available by a FDRSOS VARYON or LABEL operation.

Action: Correct the jobstream and resubmit the job.

CONTINUED. . .

230.22 CONTINUED. . .

**FDR042 RESTORE FROM BACKUP OF VOL=vvvvvv UNIT=uuuu CREATED ON
DATE=yy.ddd TIME=hh.mm.ss**

Reason: Documents the creation time and date of the backup from which a restore is being done, as well as the volume serial (vvvvvv) and original device address (uuuu) of the disk that was dumped.

**FDR090 DIAGNOSTIC MINI DUMP-FDRDMPRT VER v.r/nnb-INNOVATION DATA
PROCESSING DATE/TIME – yyddd/hh.mm.ss PAGE -- nn**

Reason: General page heading for the MINI DUMP processor listing the version, level, date, time and page.

FDR091 REGS

Reason: General register contents formatted by the MINI DUMP processor.

FDR092 type LEN=nnnn(hhhh) LOC=lllll

Reason: Control block formatted by the MINI DUMP processor. 'type' is the name of the control block, 'nnnn' is the length in decimal, 'hhhh' is the length in hex, and 'lllll' is the storage location in hex.

FDR093 CCWS LEN=nnnn(hhhh)

Reason: CCWs formatted by the MINI DUMP processor.

FDR094 reason – CCWS NOT PRINTED(TERMINATED)

Reason: CCW formatting was requested in a MINI DUMP. However, for the reason printed, CCW printing was terminated.

Action: Contact Innovation technical support for further problem determination.

FDR099 NEAR RELATIVE DATA LOCATION nnn -- error description

Reason: The common parsing routine encountered an error in parsing the user specified control statements. The approximate location of the error was position nnn, counting the first position as 000.

Action: Correct error described in 'error description' and re-submit job.

CONTINUED. . .

230.22 CONTINUED . . .

FDR122 OPERATION STATISTICS FOR SOS VOLUME...volser

Reason: This is the header for a table of statistics about the dump, restore, or print operation that was performed on the Open System volume indicated. This may include:

BYTES ON VOLUME - total size, in bytes, of the Open System volume being processed

BYTES READ FROM DASD - total number of bytes actually backed up from the Open System volume (should be the same as "bytes on volume" for full backups).

BYTES WRITTEN TO DASD - total number of bytes actually restored to the Open System volume (should be the same as "bytes on volume" for full restores from full backups).

BYTES ON BACKUP - total number of bytes actually written to or read from the backup. For uncompressed backups, this will be slightly higher than the DASD bytes because of control data. For compressed backups (COMPRESS= on the DUMP statement), this is the bytes after compression.

COMPRESSION SAVINGS - percentage of the backup file size saved by FDRSOS compression (COMPRESS= on the DUMP statement). This is calculated as the "Bytes on Backup" divided by the bytes that would have been written to the backup without compression (not displayed). This will be zero unless COMPRESS=ALL was specified; compression savings from IDRC tape hardware compression is not displayed.

DASD SECTORS BACKED UP|WRITTEN - number of 512 byte data sectors read from or written to the Open System volume.

BACKUP BLOCKS WRITTEN|READ - number of physical blocks written to or read from the backup file.

DASD EXCPS - number of write/read I/O requests issued to the Open System volume.

BACKUP FILE EXCPS - number of write/read I/O requests issued to the backup file.

CPU TIME (SECONDS) - the CPU (TCB) time required to process this volume, in seconds and thousandths of a second.

ELAPSED TIME (MINUTES) - the actual time, in minutes and tenths, required to process this volume.

BACKUP|RESTORE TIME (EXCLUDING MOUNTS) - the actual time, in minutes and tenths, required to process this volume, excluding the time spent waiting for input or output tape volumes to be mounted (at OPEN or EOVS).

BACKUP COPY n ON type DSN=dsname VOL=volumes - documents the location of the backup file created. "n" is the copy number (1 or 2) and "type" is DISK or TAPE.

RESTORE FROM DSN=dsname - documents the name of the backup file being restored.

230.22 CONTINUED . . .**FDR129 I/O ERROR ON DISK PACK--LAST SEEK ADDRESS READ X'bbbbbbbbb'**

- Reason:** An I/O error or logical error was detected by FDRSOS on this volume. "bbbbbbbbb" is the sector number (in hex) of the first sector referenced by the I/O that got the error (the actual sector in error may follow that sector). An IBM IOS000I message may also have been printed on the Job log. A FDR149 message will be printed detailing the I/O error
- Action:** FDRSOS will attempt to continue processing from this point, bypassing the data in error, but this may make the backup or the restored disk unusable. If more than MAXERR= errors occur the operation will be terminated with a U0101 ABEND. Contact Innovation technical support for assistance.

**FDR149 I/O TRACE SECT|CCHH=X'xxxxxxxx' mmm.mmm MSEC ttt DD=ddname UNIT=unit
CCW/DATA - IOB iobdata
CCW/DATA - SENSE sensedata
ccwdata**

- Reason:** This message is produced for FDRSOS internal traces and is also printed for I/O errors on either the Open System Volume (in which case SECT= documents the relative sector number first referenced by the I/O operation on which the error occurred) or the backup file (CCHH= documents the disk location for backup files on DASD, and is meaningless for tape backups). The elapsed time of the I/O in milliseconds is shown as "mmm.mmm". "ttt" is an internal clock value.
- "ddname" and "unit" document which device this message relates to.
- "iobdata" documents the I/O termination status. If a unit check (device error) occurred, "sensedata" will document the sense bytes returned by the device.
- "ccwdata" documents the CCW chain that was issued, along with 8 (occasionally 16) bytes of data associated with each CCW.
- Action:** Check the job log of the FDRSOS job for IBM messages which may relate to the error. If the cause of the error is not obvious, contact Innovation tech support for assistance in interpreting and correcting the error.

FDR200 BLOCK DROPPED--(synadaf info)--(data block)

- Reason:** FDRSOS or FDRSOSTC has encountered a permanent BSAM read or write I/O error on a backup data set. The cause of the error is documented as "synadaf info" as provided by the IBM SYNAD exit. "data block" is the first 20 bytes of backup data block, if available. An IBM IOS000I I/O error message may also appear in the job's JOB LOG.
- Action:** The step will be terminated with a U0200 abend.

230.22 CONTINUED. . .**FDR203 PREMATURE TAPE END OF FILE DSN=dsname**

- Reason:** FDRSOS detected an end-of-file on the backup data set without encountering the FDR trailer identification record. Possible causes are:
- The backup that created this file did not complete successfully.
 - The backup run completed successfully, but the backup file was not correctly cataloged.
 - The JCL for the restore specifies volume serial numbers for the backup file, but the last volume has been omitted, or the volume serials are out of order.
- Action:** In the last two cases, you can do the restore by specifying the correct volume serial numbers on the DD statement for the backup file.

FDR204 TAPE BLOCK LENGTH CHECK--BLOCK BYPASSED

- Reason:** FDRSOS records the length of the block written internally. FDRSOSTC detected that the length of the block read from the input did not match the length recorded. An FDR MINI DUMP is printed. If Message FDR204 occurs many times, the cause is probably that you are copying from a tape that was created by using a utility program (such as IEBGENER) to copy an FDRSOS backup file. Most copy utilities cannot copy FDR backup tapes. FDRSOSTC or FATAR (Ver. 4.4) must be used to copy FDRSOS backups.
- Action:** FDRSOSTC will continue processing bypassing this block. Data may be lost from the tape.

FDR206 TAPE BLOCK COUNT ERROR

- Reason:** The number of blocks read from a backup tape did not match the block count in the tape trailer label. A MINI DUMP is printed of the DCB, UCB and registers. The registers are from the block count exit. Register zero (0) contains the block count from the trailer label, which reflects the number of blocks written during the backup run. The fourth word of the DCB contains the count of blocks read during the restore.
- Action:** FDRSOS will ignore the error and continue processing.

FDR210 I/O ERROR ON BACKUP DD=ddname

- Reason:** An I/O error has occurred writing the backup data set. A FDR149 message will follow to document the error.
- Action:** The backup subtask will abend immediately with a U0200 abend.

230.22 CONTINUED. . .

FDR211 FDRSOS ERROR ON DD=ddname REASON=reason

Reason: FDRSOS RESTORE had an error attempting to initialize the DD statement shown, for the reason shown. Reason codes include:

- 1 - NO MATCHING MOUNT TAPEDD - a TAPEX DD was found in the JCL but there was no matching DISKx DD or MOUNT statement to indicate where to restore the backup.
- 2 - UNIT/VOLSER NOT SPECIFIED - UNIT= and/or VOL= must be specified on the MOUNT statement.
- 3 - UNIT ADDRESS NOT FOUND - UNIT= was specified but no offline DASD device with that address was found.
- 4 - DD STATEMENT IS MISSING - TAPEDD=x was specified but no TAPEX DD statement was found.
- 5 - VOLSER NOT FOUND - VOL= was specified but that volser was not found in the UCB of any offline DASD device. You may need to execute the VARYON or LABEL function of FDRSOS.
- 6 - TAPEDD NOT SPECIFIED - TAPEDD= was not specified
- 7 - NOT A VALID SOS DEVICE - the unit or volume specified was not a valid Open System volume.

Action: FDRSOS will bypass this DD statement and continue processing.

230.22 CONTINUED. . .

FDR212 FDRSOS ERROR UNIT=uuuu REASON=reason

Reason: FDRSOS had an error attempting to initialize or process the Open System volume on the device address shown as "uuuu", for the reason shown. Reason codes include:

- 1 - VARY FAILED - VOLSER IS A DUPLICATE** - VARYON or LABEL failed because the volume serial of the Open System volume is already assigned to another Open System or S/390 volume.
- 2 - VOLSER FILLED IN BUT DOES NOT RESPOND** - a Open System UCB has a volume serial set, but the device does not respond. Check the EMC Symmetrix subsystem.
- 3 - DEVICE IS RESERVED TO ANOTHER MACHINE** - RESERVE=YES was specified (or defaulted) but the Open System volume is reserved to another system.
- 4 - NOT A RESPONDING SOS UNIT** - unit is not responding to I/O requests.
- 5 - UNIT IS DUPLICATE OR INCLUDED IN PRIOR MOUNT** - the UNIT= value specified on the MOUNT statement is a duplicate of that on a previous MOUNT (or is included in a prefix on a previous MOUNT).
- 6 - UNIT DOES NOT CONTAIN SPECIFIED VOLSER** - both UNIT= and VOL= were specified, but the volume serial was not found on any of the devices specified.
- 7 - TAPEDD=X FOR MOUNT COMMAND NOT FOUND** - TAPEDD= was omitted when required.
- 8 - ERROR READING/Writing THE LABEL SECTOR** - an I/O error occurring on the volume label from an Open System volume during a VARYON or LABEL operation.
- 9 - VOLUME DOES NOT CONTAIN A LABEL** - for a VARYON operation, the volume was not previously initialized by a LABEL operation.
- A - SETVOL MISSING OR NOT FULL UNIT ADDRESS** - for a LABEL operation, either the SETVOL= operand was omitted or UNIT= does not specify a single address.
- B - I/O ERROR ACQUIRING STATUS** - PRINT=STATUS was specified but an I/O error occurring reading the Open System volume.
- C - I/O ERROR ERASING SECTORS** - an I/O error occurring during an ERASE operation.
- D - SETVOL VOLSER CONTAINS INVALID CHARACTERS** - the volume label contained other than alphanumeric and national characters.
- E - UNIT ADDRESS IS INVALID OR NOT 4 CHARACTERS** - in a UNIT= parameter, either non-hexadecimal characters were specified, or the length was less than 4 and did not end with *. Device addresses must be 4-digit addresses.

Action: FDRSOS will bypass this Open System volume and continue processing.

FDR213 FDRSOS PSEUDO MOUNTED UNIT=uuuu VOL=volser - SYM-NUMBER=X'ssss-ssss'

Reason: FDRSOS successfully processed a VARYON statement for the Open System volume on device address "uuuu". Its volume serial is "volser" and its internal EMS Symmetrix ID is "ssss-ssss" (subsystem number and internal device number).

230.22 CONTINUED. . .

FDR214 FDRSOS PSEUDO DIS-MOUNTED UNIT=uuuu VOL=volser

Reason: FDRSOS successfully processed a VARYOFF statement for the Open System volume on device address "uuuu". Its volume serial was "volser".

FDR215 FDRSOS LABELED UNIT=uuuu TO VOL=volser

Reason: FDRSOS successfully processed a LABEL statement for the Open System volume on device address "uuuu". Its volume serial has been set to "volser".

**FDR216 STATUS OF SOS SCSI VOLUME UNIT=uuuu VOL=volser - SYM-NUMBER=X'ssss-ssss'
UNIT=uuuu status message
UNIT=uuuu IS UNKNOWN VOLUME TYPE**

Reason: If PRINT=STATUS is specified on a FDRSOS control statement, FDRSOS will attempt to determine the type of each Open System volume processed. A block of FDR216 messages are printed for each volume, identified by its device address "uuuu" and volume serial "volser" and internal EMC Symmetrix ID "ssss-ssss" (subsystem number and internal device number). The header line shown will always appear, following by one or more additional FDR216 messages. For a RESTORE, this information is printed for the volume both before and after the restore. This information may be useful to you to confirm which Open System volumes correspond to which S/390 device addresses. When the volume type is successfully identified, FDRSOS will show the type of Open System which created the volume (e.g., IBM AIX, HP-UX, SUN SOLARIS, NOVELL, OS/2, etc.) plus additional information about the contents of the volume (volume or volume group identifications, volume group information, UNIX mount points, and other information; the details will vary depending on the type of Open system). If "UNKNOWN VOLUME TYPE" is printed, FDRSOS was unable to identify the type of Open System volume. However, FDRSOS is not sensitive to the format of the volume and will still be able to process it successfully.

NOTE: The volume information displayed by FDRSOS is based on Innovation's current understanding of the format of volumes created by each of the Open Systems supported, as obtained by research and experimentation. Innovation will update the PRINT=STATUS display in subsequent releases of FDRSOS based on experience. If PRINT=STATUS does not display your Open System volumes correctly, please contact Innovation so that we can obtain information necessary to improve the displays.

230.22 CONTINUED. . .

FDR219 FDRSOS UNIT=uuuu IS A BCV VOLUME - STATUS=X'status bytes'

Reason: During a VARYON operation, the device identified by the device address ("uuuu") was identified as a EMC Symmetrix BCV (Business Continuance Volume). The "status bytes" are 16 hex bytes of status from a special QUERY to the BCV volume. If the second or third word of the status is non-zero, then the BCV volume is not yet completely synchronized with the primary volume.

ERROR UPDATING VOLINFO REASON=reason.**FDR220**

Reason: During a RESTORE with VOLRESET=NO, FDRSOS encountered an error attempting to preserve all of the volume ID information on the output Open System volume. "Reason" documents the error:

1 - AIX MAXIMUM LOGICAL VOLS EXCEEDED - while attempting to update the logical volumes on an IBM AIX/6000 volume group, more than 255 logical volumes were encountered. This probably indicates that the Open System volume, although it appears to be formatted by AIX, is not properly formatted or has been overlaid.

FDR221 TAPEOUT SYNAD ERROR, DCB ADDRESS UNIDENTIFIED.

Reason: FDRSOSTC entered its SYNAD routine for TAPEOUT and the DCB passed was not the expected value. This is an internal error and should not occur.

Action: A U0200 ABEND will be issued with Reason Code 221 (in register 15). Examine Joblog and Syslog for any messages that may be the cause of the error on the TAPEOUT device. Correct these errors and resubmit. If necessary, contact Innovation Technical Support for assistance.

230.22 CONTINUED . . .**FDR302 CONTROL STATEMENT ERROR NEAR REL LOCATION nn -- REASON x -- JOB TERMINATED**

Reason: An error was encountered during the processing of a user-supplied control statement. If "NEAR REL LOCATION nn" appears, the keyword or operand causing the error is at or near column "nn" on the input statement. The single character reason code defines the error:

- 1 – A SELECT/EXCLUDE statement did not specify any operands. Control statement was blank after the Command name.
- 2 – Command on the first control statement was incorrectly specified; i.e.: was not DUMP, RESTORE, PRINT, ERASE, LABEL, VARYON, or VARYOFF.
- 3 – Operand on the first control statement was incorrectly specified
- 4 – Operand did not end with a blank or comma.
- 5 – SYSIN data set was empty.
- 6 – Expected continuation statement was not found. The previous statement ended with a comma and a blank.
- 7 – Invalid or incompatible operands were specified or a required operand was omitted.
- 8 – An operand on a SELECT/EXCLUDE statement specified a blank or comma after the equal sign.
- 9 – One of the operand values exceeded the maximum permitted number of characters or digits.
- F – An operand requiring a numeric value had non-numeric characters.
- G – An operand requiring a numeric value had too many digits.
- I – Keyword is invalid under the operation indicated.
- J – Control statement was completely blank.
- K – A required operand such as STARTBLK= was not specified on the preceding SELECT/EXCLUDE.
- Q – Keyword exceeded maximum value or was negative.
- R – TYPE=xxx was specified multiple times.
- S – An operand on the SELECT-type Command was specified multiple times or was mutually exclusive with another operand.
- U – The TYPE=xxx operand was missing or invalid on the DUMP, RESTORE, ERASE or PRINT statement.
- V – The FROMBLK= value was greater than the TOBLK= value.

Action: The FDRSOS step is terminated; no further control statements are read. Correct error and resubmit job.

FDR303 CARD IMAGE--control statement image PARM ENTRY

Reason: Display all input control statements from the SYSIN data set. If 'PARM ENTRY' appears after the control statement, entry was passed from the PARM FIELD on the EXEC statement.

230.22 CONTINUED . . .**FDR316 FDR DID NOT FIND REQUESTED item**

- Reason:** This message is displayed when the statement identified by "item" did was not processed. This could occur when:
- 1) a MOUNT did not select any Open System volumes, perhaps because the UNIT= and/or VOL= operands were incorrectly specified or the volume specified by VOL= has not been mounted by a VARYON statement.
 - 2) a SELECT/EXCLUDE statement specified a TAPEDD= or VOL= which did not match any volume being processed.
- Action:** Be sure that you specified 4-digit device addresses (for example, UNIT=01E0 or UNIT=01F*) even if your operating system supports only 3-digit addresses; add a leading zero if necessary. Verify that the UNIT=, TAPEDD=, or VOL= specifies values appropriate for your configuration and jobstream.

FDR319 FDR OPERATION ABNORMALLY TERMINATED COMP CODE=Ssss Uuuuu

- Reason:** A FDRSOS dump or restore subtask has abnormally terminated with either a system ABEND code of "sss" or a user ABEND code of "uuuu" (which ever is non-zero). Check IBM documentation for the meaning of system ABENDs.
- Action:** FDRSOS will terminate this Open System volume and continue processing.
If your JCL includes a SYSUDUMP or SYSABEND DD statement, a diagnostic dump will be produced in most cases. Check for other FDR messages which may be printed prior to the ABEND; these may help you understand the error. Check IBM documentation for the meaning of system ABENDs, [see Section 230.23](#) for the meaning of user ABENDs. If you are unable to determine the cause of the ABEND, contact Innovation tech support; have all messages and the diagnostic dump available.

FDR324 DDNAME=ddname MISSING OR IN ERROR REASON x

- Reason:** The DD statement identified by "ddname" had an error identified by the single character reason code:
- 1 – The SYSIN DD statement was missing or could not be opened. A U0401 ABEND will occur.
 - E – The "DISKx" DD statement appeared more than once.
 - F – DISP=MOD was specified on a TAPEX or TAPEXX DD statement. This may also mean that you directed multiple backups to the same TAPEX DD by the same TAPEDD=x on multiple MOUNT statements.
 - G – Both TAPEXX and TAPEXX point to backup data sets on disk. You cannot create 2 simultaneous disk backups. You can create simultaneous backups on tape and disk.

230.22 CONTINUED . . .**FDR336 DYNAMIC ALLOCATION ERROR COMP=cc, CODE=nnnn, INFO=iiii, DDNAME=ddname**

Reason: Dynamic allocation was requested for a disk volume or a data set (such as a disk or tape backup data set), but the allocation failed. "cc" is the return code in REG 15, "nnnn" and "iiii" are from the dynamic allocation parameter list. These are documented in the IBM manuals SPL: System Macros and Facilities for MVS/XA, and Authorized Assembler Services Guide for MVS/ESA and OS/390. Frequently encountered CODE values include:

210– requested data set not available (e.g., another job had the required backup data set allocated with DISP=OLD).

220– requested volume not available (e.g., another job is using a tape volume required).

484– the operator replied "NO" to Message IEF235D.

1708– data set not found in the catalog (e.g., BACKUPDSN= was misspelled).

Action: This volume or data set operation will be bypassed. If necessary, contact INNOVATION for assistance.

FDR370 OPEN ERROR -- TAPE BYPASSED DSN=dsname

Reason: An error occurred opening the backup data set for a dump or restore.

Action: The operation is bypassed. Check for IBM messages in job log. If necessary, contact Innovation tech support for assistance.

FDR390 NO SECTORS MET SELECTION CRITERIA

Reason: A TYPE=PARTIAL operation was executed, but the range of data specified by FROMBLK= and TOBLK= did not exist on the Open System volume processed. The values specified were probably larger than the size of the volume.

230.22 CONTINUED . . .**FDR400 FDR SAFEGUARD OPEN STORAGE -- FDRSOSTC VER v.r/mmb
--INNOVATION DATA PROCESSING DATE=yy.ddd PAGE nnn**

Reason: This is the FDRSOSTC page heading, specifying the version level of FDRSOSTC. "v.r" indicates the version and release (e.g., 5.3), "mm" is a 2-digit number indicating the maintenance level and "b" will be "P" for a production version, "T" for a trial, and "B" for a beta test version.

FDR402 INVALID CONTINUATION

Reason: FDRSOSTC detected that the last control statement ended with a comma but no continuation statement was found.

Action: Correct and resubmit job.

FDR446 1ST RECORD NOT FDR HEADER

Reason: The first record read from the backup data set was not the expected special FDRSOS header record.

Action: A U0626 ABEND will be issued. Check that the input data set is a FDRSOS backup and resubmit. Contact Innovation Technical Support for assistance if the problem recurs.

FDR447 CONTROL RECORD NOT FOUND

Reason: Each set of backup records is preceded by a control record describing them. A control record was not found when it was expected. This may be due to tape errors.

Action: A U0626 ABEND will be issued. Contact Innovation Technical Support for assistance.

**FDR450 TAPEIN SUCCESSFULLY COPIED TO TAPEOUT--BLOCKS READ nnnnnn BLOCKS
WRITTEN nnnnnn**

Reason: One file has been successfully copied from TAPEIN to TAPEOUT (and optionally to TAPE2OUT as well) by FDRSOSTC. Blocks read and blocks written should be the same. This message will be followed by Message FDR615 detailing the data sets read and written.

FDR451 BLOCK GREATER THAN 32760 TO OUTPUT ON DISK ...JOB TERMINATED

Reason: FDRSOSTC detected the input file was on tape and blocked at over 32k and the output file to TAPEOUT is on Disk (Initial release of FDRSOSTC does not support this function. A future release will eliminate this restriction)

Action: A U0502 ABEND is issued. Contact INNOVATION for assistance after obtaining a storage dump.

FDR465 function FAILED ON DDNAME=ddname-- MISSING OR MISSPECIFIED.

Reason: a) ddname printed is missing or misspelled.
b) OPEN failure occurred for ddname specified in message.
c) READ JFCB failure occurred for ddname specified in message.

Action: A U0629 ABEND occurs. Correct the specified DDNAME and resubmit the job.

230.22 CONTINUED . . .**FDR484 INTERNAL LOGIC ERROR--JOB TERMINATED**

Reason: FDRSOSTC has encountered an unexpected error condition.

Action: If this message is accompanied by another error message see that message for appropriate action. Otherwise insure the SYSUDUMP dd was included in the job and obtain a storage dump then call Innovation Technical Support.

FDR498 program (version) PROCESSING COMPLETED WITH ERRORS

Reason: The program listed in the message completed the required processing but recoverable errors were encountered.

Action: Check the output, correct and rerun or restart as appropriate.

FDR499 program (version) PROCESSING COMPLETED

Reason: The program listed in the message completed the required processing.

FDR516 Function REGISTERS -- R0=xxxxxxx R1=xxxxxxx R15=xxxxxxx

Reason: A CAMLST catalog request or an OBTAIN request failed. The type of CAMLST function and the registers make diagnosing the problem relatively simple. The return codes from catalog management are documented in the IBM manuals SPL: DATA MANAGEMENT for MVS or CATALOG ADMINISTRATION GUIDE for MVS/XA and MVS/ESA.

Action: Check the return code(s) and take corrective action if the error is apparent or call INNOVATION technical support for assistance.

230.22 CONTINUED . . .**FDR604 ERROR OCCURED POSITIONING TAPEIN REASON CODE-x**

Reason: FDRSOSTC detected an error when positioning the input tape. The reason code x may be:

REASON	EXPLANATION
1	Tape mark not read when expected.
2	Tape mark read when not expected.
3	Trailer label read is not EOVI/EOF1.
4	Header label read is not HDR1.
5	Tape drive unit check during positioning.
6	Other I/O error during positioning.

Action: A U0659 ABEND follows. This may be due to hardware problems with the tape drive. If it occurs with the same input tape on more than one tape drive, contact INNOVATION for technical support.

FDR615 dddddddd DSN=dsname FILE=fn VOL=vvvvvv...

Reason: Printed by FDRSOSTC at the end of each file copied. One FDR615 message will be printed for TAPEIN, TAPEOUT, and TAPE2OUT (if present). It documents the input or output DDNAME, data set name, file number (0 if on disk), and up to 20 volume serials.

FDR997 FDR ABNORMALLY TERMINATED VOL=vvvvvv

Reason: This FDRSOS subtask for the volume indicated has encountered an error from which it cannot recover. Other messages may be printed to detail the error.

Action: The subtask will terminate with a user ABEND; [see Section 230.23](#) to interpret the abend code. If necessary, contact Innovation tech support for assistance.

FDR998 FDR DUMP|RESTORE COMPLETED WITH ERRORS VOL=vvvvvv

Reason: This FDRSOS subtask for the volume indicated ran to completion but errors occurred during the operation. Other messages may be printed to detail the error.

Action: Review the messages to determine if corrective action is necessary. If necessary, contact Innovation tech support for assistance.

FDR998 FDR COMPLETED WITH ERRORS

Reason: The FDRSOS step had one or more errors. Other messages may be printed to detail the errors. The step completion code will be set to 12.

Action: Review the messages to determine if corrective action is necessary. If necessary, contact Innovation tech support for assistance.

FDR999 FDR SUCCESSFULLY COMPLETED

Reason: The FDRSOS step completed without errors. The step completion code will be set to 0 (zero).

230.23 FDRSOS ABEND CODES

FDRSOS or its utility programs may abend with any of the following user abend codes. In many cases, a diagnostic message is printed before the abend, so look up any error messages that were printed first. If no message was printed that relates to this abend, then read the explanation below. Call Innovation technical support if you need assistance understanding or resolving the error.

U0100	Open Error Trying to Open a DASD or Tape DCB Usually preceded by a FDR324 message. Check the job log for IBM messages which may indicate the reason for the error.
U0101	Maximum I/O Errors Exceeded on a Direct-Access Device More disk I/O errors than are allowed by the MAXERR= operand occurred. If you want to complete the operation in spite of the errors, specify a larger MAXERR= value. However, many data sets may be invalid.
U0200	I/O Error on Backup Data Set A permanent I/O has occurred on the backup tape or disk data set. The backup is not usable.
U0201	Premature End-of-File on Input The end of a backup file was reached without finding the special FDR trailer record.
U0204	Invalid Block read An invalidly formatted block was read from a backup.
U0205	Not an FDRSOS Backup A restore detected that the backup was not created FDRSOS.
U0210	Internal error in Compress.
U0401	SYSIN DD Statement Error SYSIN DD statement missing or incorrectly specified or I/O error on SYSIN data set.
U0402	SYSPRINT/SYSPRINn DD Statement Error One of the above DD statements is missing or incorrectly coded or I/O error occurred processing the data set. There must always be SYSPRINT DD statement, and there must be a SYSPRINn DD for each TAPEX DD if MAXTASKS=n was specified
U0502	One or more Control Statements are in Error
U0600	Required DD Statement is Missing or in Error
U0626	An FDR/DSF/ABR Tape Format is in Error
U0659	Internal logic error
U0660	Stack storage management error An internal error occurred.
U0801	Restore Cancelled by Operator or security failure
U0888	Copy completed with errors Errors occurred during a FDRSOSTC step but they were not severe enough to cause the entire COPY to be terminated. The listing should be examined to see what errors occurred.

READER'S COMMENT FORM
FDRSOS USERS MANUAL V5.3
INNOVATION DATA PROCESSING

If you have comments on this manual, including:

- errors in the text or typographical errors
- clarity
- suggestions for improvement in the manual
- suggestions for improvement in the product

Please complete this form and fax it to Innovation at 973-890-7147 (in Europe you may fax it to your local Innovation office as shown on the front page of the manual). You may also e-mail your comments to Innovation at 76322.2076@compuserve.com (be sure to identify the manual name in the message).

Your name: _____

Company name: _____

Mailing address: _____

E-mail address: _____

Comments: _____

CONTINUED . . .

THIS PAGE WAS INTENTIONALLY LEFT BLANK

A. FDRSOS - Windows NT Drive Locking

A.1 Introduction

FDRSOS includes a facility where a Windows NT attached EMC drive (one or more) can be locked throughout an FDRSOS backup or restore. This facility also assures that if the operating system definition of the drive is changed because you have restored a different drive to it, the operating system information about the drive is refreshed.

This facility is useful when implementing FDRSOS during:

- A backup when you wish to assure that the drive(s) are not being accessed and you wish to guarantee the integrity of the data
- A restore when you wish to assure that no users are going to be affected during the restore process.
- A restore when you are restoring an older version of an entire drive (or drives) or when you are moving the backed up contents of a drive to a different drive and wish the operating system to be notified without having to shut down the operating system.

The drive can remain locked until:

- A user presses a key.
- A predefined timeout occurs.
- Another program is run that tells the first program to unlock the drive.

Note that when restoring a drive to a different location, the drives must be of the same size.

There are two programs in this facility: UNMOUNT and REMOUNT.

Note: During a restore, UNMOUNT must be running and the drive must be locked to assure that the operating system definitions for the drive are reset. If the program is abnormally closed, the drive will be unlocked.

A.1.1 Installation

The FDRSOS disk consists of only a few files. Merely create a directory and copy the files into them:

```
md \sos
copy a:\nt\*. * c:\sos\*. *
```

or for a CD installation:

```
md\sos
copy d:\sos\nt\*. * c:\sos\*. *
```


The files are (relative to the root of the diskette or the \SOS directory on the CD):

<u>File Name</u>	<u>Description</u>
GENDATA.EXE	A test data generation program provided to help you generate a large amount of data quickly.
NT\REMOUNT.EXE	Allows you to, from a separate program at a later time, unlock a previously locked set of drives.
NT\UNMOUNT.EXE	Allows you to lock and dismount one or more drives.
NT\UNMOUNTX.EXE	An internal program, not to be called directly.

**FDRSOS Windows NT Drive Locking
Diskette Contents**

The newest version of all Innovation Data Processing workstation/server software is available on the Innovation BBS. It is a standard ANSI BBS (8-bits, 1 stop bit, no parity, 28Kbps), phone number: (201) 812-7385.

A.2 UNMOUNT.EXE

This is the program that can lock one or more drives and then forces an operating system refresh of the drive information when it is unlocked. It is a 32-bit Console program. If you wish this program can be run as a service; contact Innovation Data Processing for more information.

Unless you specify the /p option, UNMOUNT must be left running during the backup or restore to assure that the drive is locked. During testing you should verify that the drive is locked during SOS operations by attempting an access to the drive on the Windows NT machine (with a DIR command for example) and you should get an "Access is denied" error.

This program will fail if **ANY** programs are accessing the drive. You must assure that all users and programs have closed all open files on the drive to be locked before running the program. The program will write an error to the screen and return a non-zero program return code if there are any errors.

Run this program from an MS-DOS prompt or a batch file. It's format is:

```
UNMOUNT <drive letter:> ... [/n<name>] [/k] [/w<time>] [/p[<pgm>]]
```

Where:

- **<drive letter:>** is a list of one or more space separated drives to be locked. For example: **C:** or **E: F: G:**. You must specify at least one drive letter.
- **/n<name>** is an optional parameter which allows you to specify the semaphore name used when running REMOUNT. You only need to specify this parameter if you will be running REMOUNT and you wish to run, simultaneously, more than one copy of UNMOUNT.EXE at a time. For example: **/nREMOUNT2**.
- **/k** is an optional parameter that locks the drive until the user presses a key.
- **/w<time>** is an optional parameter that locks the drive for a given number of seconds. For example, for 5 minutes, specify **/w300**.
- **/p[<pgm>]** is an optional parameter that causes UNMOUNT to terminate when the drive is locked. The drive will remain locked until REMOUNT is run. The optional program parameter (<pgm>) allows you to specify a different program name than UNMOUNTX; for example, to specify UNMOUNTX on the D: drive, specify **/pD:\UNMOUNTX.EXE**. Most user will not use the optional <pgm> parameter and will specify **/p** alone.

The /p option is particularly useful, as UNMOUNT will return a program return code which can indicate to you whether the drive was successfully locked.

If you wish to lock the drives D, E and F until REMOUNT is run, specify:

```
UNMOUNT D: E: F:
```

If you wish to lock the drive D: until a user presses a key, specify:

```
UNMOUNT D: /k
```

If you wish to lock D and E, but unlock them with REMOUNT at separate times, specify:

```
UNMOUNT D: /nDDRIVE
UNMOUNT E: /nEDRIVE
```

If you wish to create a batch file that unmounts the C: drive, checks the return code and returns, it might look like:

```
UNMOUNT C: /p
if ERRORLEVEL 1 goto ERROR
Echo Successful lock - can begin SOS backup
goto SOS
:ERROR
Echo Unsuccessful lock - perform the SOS backup later
goto SOSLATER
...
```

A.3 REMOUNT.EXE

Run this program when you have run UNMOUNT and wish UNMOUNT to unlock the drive and refresh the operating system's view of it. It must be run in a separate MS-DOS prompt session as UNMOUNT.

Its form is:

```
REMOUNT [/nname]
```

Where:

- **/n<name>** is an optional parameter and is the same name specified when UNMOUNT was run.

In the example above, where UNMOUNT was run to lock D: E: F:, to unlock the drive, run:

```
REMOUNT
```

If the example above where you wish to unmount the drives separately run:

```
REMOUNT /nDDRIVE  
REMOUNT /nEDRIVE
```

Both UNMOUNT and REMOUNT return 1 if they fail and 0 if they succeed.

A.4 GENDATA.EXE

GENDATA is a bound program (DOS and OS/2) which allows you to quickly create a very large file (up to 4GB) and verify it's integrity. It's form is:

```
GENDATA <file name> <size in bytes> [/f[fill value]] [/r]
```

Where:

- **<file name>** is the fully qualified name of the file you wish to create or verify.
- **<size in bytes>** is the size of the file that you wish to create or verify.
- **/f[fill value]** is an optional parameter that allows you to be able to specify a single character to repeatedly place in the file. The fill value can be any number from 0 to 255. If you do not specify the /f option, then GENDATA will create the file with 4 byte numbers indicating the long word position. Specifying the /f option is faster than omitting it.
- **/r** causes GENDATA to verify the expected contents of a file. This option is generally used after a restore to verify that the data was replaced correctly.

For example, to create a 20,000,000 byte file named C:\TEST.DAT, specify:

```
GENDATA C:\TEST.DAT 20000000
```

To create a 1,000,000,000 byte file quickly named D:\TEST2.DAT, specify:

```
GENDATA D:\TEST2.DAT 1000000000 /f0
```

...and to verify the same file was restored correctly:

```
GENDATA D:\TEST2.DAT 1000000000 /f0 /r
```

B. FDRSOS - NetWare Drive Management

B.1 Introduction

FDRSOS includes a facility where Novell NetWare attached EMC drives (or any type of drive) can be mounted/dismounted throughout an FDRSOS backup or restore. While you can always manually mount/dismount a volume from the system console, this facility allows the process to be automated from a workstation and/or from the server. This facility also allows a workstation to load or unload an NLM and run an NCF server batch file on NetWare v4.x servers.

This facility is useful when implementing FDRSOS during:

- A backup when you wish to assure that the drive(s) are not being accessed.
- A restore when you wish to assure that no users are going to be affected during the restore process.

There is an NLM which allows a mount or dismount without user intervention (regardless of open files): NWMOUNT.NLM and a workstation program (available for Windows NT and OS/2 at this time and other operating systems in the future) NWSERVER which allows you to run server functions from a workstation.

B.1.1 Programs

The FDRSOS NetWare disk consists of only a few files in directories appropriate to the operating system they are to be run in. The files are (relative to the root of the diskette or the \SOS directory on the CD-ROM):

<u>File Name</u>	<u>Description</u>
GENDATA.EXE	A test data generation program provided to help you generate a large amount of data quickly.
NOVELL\NLM\NWMOUNT.NLM	Allows you to mount or dismount from the server a volume without user prompts. This file must be installed in the server's SYS:SYSTEM directory or some directory in the SEARCH path.
NOVELL\NLM\VOLLIB.NLM	A Novell supplied library which loads facilities required by NWMOUNT.

<u>File Name</u>	<u>Description</u>
NOVELL\NT\NWSERVER.EXE	A 32-bit Windows NT program which allows NLMs and NCF files to be executed as well as volumes mounted and dismounted on NetWare v4.x servers.
NOVELL\OS2\NWSERVER.EXE	A 32-bit OS/2 program which allows NLMs and NCF files to be executed as well as volumes mounted and dismounted on NetWare v4.x servers.

**FDRSOS NetWare Drive Management
Files**

B.2 NWMOUNT.EXE

This is a VERY DANGEROUS program which allows, from the console, the ability to mount or dismount a volume without any user prompts. If you run the server console command DISMOUNT <volume> and there are open files, you will be prompted and given an opportunity to not dismount the volume at this time. NWMOUNT allows you to dismount the volume without the prompt, facilitating automation.

Run this program from the system console (or from a workstation using NWSERVER described below). VOL-LIB.NLM must be loaded before NWSERVER. It's format is:

```
LOAD NWMOUNT <volume> [/Mount] [/Dismount]
```

Where:

- **<volume>** Is the NetWare server's volume name that you wish to mount or dismount. Do not include the trailing colon.
- **/Mount** requests that the specified volume be mounted. You should be sure that the SOS backup/restore has completed before performing this function.
- **/Dismount** requests that the specified volume be dismounted. As stated above, this is very dangerous.

For example, if you wish to dismount the VOL1 volume from the server console, specify:

```
LOAD NWMOUNT VOL1 /Dismount
```

If you wish to remount the volume, specify:

```
LOAD NWMOUNT VOL1 /Mount
```

Note that if you have dismounted the SYS volume, you can not run NWMOUNT to remount it as the NWMOUNT.NLM is not available (it is on an volume that is currently unavailable). However, you can use NWSERVER (described below) to remount the SYS volume from a workstation.

NWMOUNT creates a separate screen to display messages in. If there is an error, this screen remains displayed until a key is pressed to give you an opportunity to read the messages. All significant messages are also written to the system console (for console logging purposes).

There are other advanced switches including:

- **/List** allows you to list all the volumes on the server (whether mounted or not, whether valid or not) with a significant amount of detail. Do not specify <volume> with this option.
- **/One** allows you to list volume detail for a single, specified volume.
- **/F<file name>** allows all output to be written to the specified file name instead of the screen. The file name should be specified in NetWare format immediately after the /F option (for example: /FSYS:NWMOUNT.OUT). This option can be combined with any other option.

- **/R<new volume name>** allows you to rename <volume> to <new volume name>. Since this option does not interact with directory services, and since there are problems renaming a volume which has a duplicate volume name, we recommend using this option only after consultation with Innovation Technical Support.

B.3 NWSERVER.EXE

NWSERVER.EXE is a Windows NT or OS/2 program which allows you to perform a number of server related functions from a workstation.

Its form is:

```
NWSERVER <server> <function> ...parameters... [/t]
```

Where:

- **<server>** Is the NetWare server name that you wish to access. This must always be the first parameter and is required.
- **<function>** One of the function types below. Subsequent parameters are specific to the function type.
- **[/t]** An optional parameter that you can specify after the last function parameter to enable tracing.

The functions supported are:

- **NCF** Specify this function if you wish to run an NCF batch file on the server. The single required parameter is the fully qualified path name of the NCF file that you wish to run. For example:

```
NWSERVER SERVER1 NCF SYS:SYSTEM\TEST.NCF
```

- **NLM** Specify this function if you wish to load an NLM on the server. The single required parameter is the fully qualified path name of the NLM that you wish to load. Note that you can not pass parameters to the NLM. For example:

```
NWSERVER SERVER1 NLM SYS:SYSTEM\USNDS.NLM
```

- **UNLOAD** Specify this function if you wish to unload a pre-loaded NLM on the server. The single required parameter is the name of the NLM that you wish to unload (this parameter is not usually fully qualified). For example:

```
NWSERVER SERVER1 UNLOAD USNDS.NLM
```

- **MOUNT** Specify this function if you wish to mount a specified volume. The single required parameter is the volume name (without the trailing colon). This function is particularly useful for automating the mounting of the SYS volume when you used NWMOUNT to dismount it. For example:

```
NWSERVER SERVER1 MOUNT SYS
```

- **DISMOUNT** Specify this function if you wish to dismount a specified volume. The single required parameter is the volume name (without the trailing colon). Note that this function is not terribly useful, as if there are ANY open files on the volume, the console will display a list of these files and wait for user intervention to acknowledge the breaking of the connection to these files. For example:

```
NWSERVER SERVER1 DISMOUNT VOL1
```

Note that all NWSERVER functions require that you be logged in with console operator rights.

NWSERVER returns a zero program return code if the function could be initiated. For MOUNT, DISMOUNT and UNLOAD, this indicates whether the actual function was successful. For NCF, and NLM this indicates whether the function was started, but not the final return code.

B.4 Automation

To automate the process of dismounting, a volume for backup most users will create two batch files on their workstation (one for mounting and one for dismounting), and one NCF batch file on their server (for dismounting).

For example, through host automation which can run an application on a workstation (FDR/UPSTREAM is such a facility) you wish to run a job which will dismount a volume before beginning an EMC backup. Create a batch file on the workstation similar to the following:

```
Echo This batch file DISMOUNTs VOL1 on SERVER1
Echo It should only be called from the host.
NWSERVER SERVER1 NCF SYS:SYSTEM\VOL1DOWN.NCF
```

Sample batch file to dismount a volume

Note that your host facility should check the program return code from NWSERVER to see if the NCF file was correctly executed.

You would then write an NCF file on the server which would unmount the volume:

```
NWMOUNT VOL1 /Dismount
```

Sample VOL1DOWN.NCF

When the SOS backup has completed, run a batch file to remount the dismounted volume:

```
Echo This batch file remounts VOL1 on SERVER1
NWSERVER SERVER1 MOUNT VOL1
```

Sample batch file to remount a volume

Again, you will want to check the program return code to verify that the mount was successful; if not you will want to take manual action to examine the problem.

B.5 GENDATA.EXE

GENDATA is a bound program (DOS and OS/2) which allows you to quickly create a very large file (up to 4GB) and verify it's integrity. It's form is:

```
GENDATA <file name> <size in bytes> [/f[fill value]] [/r]
```

Where:

- **<file name>** is the fully qualified name of the file you wish to create or verify.
- **<size in bytes>** is the size of the file that you wish to create or verify.
- **/f[fill value]** is an optional parameter that allows you to be able to specify a single character to repeatedly place in the file. The fill value can be any number from 0 to 255. If you do not specify the /f option, then GENDATA will create the file with 4 byte numbers indicating the long word position. Specifying the /f option is faster than omitting it.
- **/r** causes GENDATA to verify the expected contents of a file. This option is generally used after a restore to verify that the data was replaced correctly.

For example, to create a 20,000,000 byte file named C:\TEST.DAT, specify:

```
GENDATA C:\TEST.DAT 20000000
```

To create a 1,000,000,000 byte file quickly named D:\TEST2.DAT, specify:

```
GENDATA D:\TEST2.DAT 1000000000 /f0
```

...and to verify the same file was restored correctly:

```
GENDATA D:\TEST2.DAT 1000000000 /f0 /r
```

C. FDRSOS - OS/2 Drive Locking

C.1 Introduction

FDRSOS includes a facility where an OS/2 attached EMC drive (one or more) can be locked throughout an FDRSOS backup or restore. This facility also assures that if the operating system definition of the drive is changed because you have restored a different drive to it, the operating system information about the drive is refreshed.

This facility is useful when implementing FDRSOS during:

- A backup when you wish to assure that the drive(s) are not being accessed and you wish to guarantee the integrity of the data
- A restore when you wish to assure that no users are going to be affected during the restore process.
- A restore when you are restoring an older version of an entire drive (or drives) or when you are moving the backed up contents of a drive to a different drive and wish the operating system to be notified without having to shut down the operating system.

The drive can remain locked until:

- A user presses a key.
- A predefined timeout occurs.
- Another program is run that tells the first program to unlock the drive.

Note that when restoring a drive to a different location, the drives must be of the same size.

There are two programs in this facility: UNMOUNT and REMOUNT.

Note: During a restore, UNMOUNT must be running and the drive must be locked to assure that the operating system definitions for the drive are reset. If the program is abnormally closed, the drive will remain locked and the system must be rebooted to make it accessible.

C.1.1 Installation

The FDRSOS Windows NT disk consists of only a few files. Merely create a directory and copy the files into them:

```
md \sos
copy a:\os2\*. * c:\sos\*. *
```

The files are (when seen from the root of a diskette or the \SOS directory on the CD):

<u>File Name</u>	<u>Description</u>
GENDATA.EXE	A test data generation program provided to help you generate a large amount of data quickly.
OS2\REMOUNT.EXE	Allows you to, from a separate program at a later time, unlock a previously locked set of drives.
OS2\UNMOUNT.EXE	Allows you to lock and dismount one or more drives.
OS2\UNMOUNTX.EXE	An internal program, not to be called directly.

**FDRSOS OS/2 Drive Locking
Diskette Contents**

The newest version of all Innovation Data Processing workstation/server software is available on the Innovation BBS. It is a standard ANSI BBS (8-bits, 1 stop bit, no parity, 28Kbps), phone number: (201) 812-7385.

C.2 UNMOUNT.EXE

This is the program that can lock one or more drives and then forces an operating system refresh of the drive information when it is unlocked. It is a 32-bit OS/2 full-screen, windowable program.

Unless you specify the /p option, UNMOUNT must be left running during the backup or restore to assure that the drive is locked. During testing you should verify that the drive is locked during SOS operations by attempting an access to the drive on the OS/2 machine (with a DIR command for example) and you should get a system pop-up message indicating the the drive is locked.

This program will fail if **ANY** programs are accessing the drive. You must assure that all users and programs have closed all open files on the drive to be locked before running the program. The program will write an error to the screen and return a non-zero program return code if there are any errors.

Run this program from an OS/2 full-screen or Window. It's format is:

```
UNMOUNT <drive letter:> ... [/n<name>] [/k] [/w<time>] [/p[<pgm>]]
```

Where:

- **<drive letter:>** is a list of one or more space separated drives to be locked. For example: **C:** or **E: F: G:**. You must specify at least one drive letter.
- **/n<name>** is an optional parameter which allows you to specify the semaphore name used when running REMOUNT. You only need to specify this parameter if you will be running REMOUNT and you wish to run, simultaneously, more than one copy of UNMOUNT.EXE at a time. For example: **/nREMOUNT2**.
- **/k** is an optional parameter that locks the drive until the user presses a key.
- **/w<time>** is an optional parameter that locks the drive for a given number of seconds. For example, for 5 minutes, specify **/w300**.
- **/p[<pgm>]** is an optional parameter that causes UNMOUNT to terminate when the drive is locked. The drive will remain locked until REMOUNT is run. The optional program parameter (<pgm>) allows you to specify a different program name than UNMOUNTX; for example, to specify UNMOUNTX on the D: drive, specify **/pD:\UNMOUNTX.EXE**. Most user will not use the optional <pgm> parameter and will specify **/p** alone.

The /p option is particularly useful, as UNMOUNT will return a program return code which can indicate to you whether the drive was successfully locked.

If you wish to lock the drives D, E and F until REMOUNT is run, specify:

```
UNMOUNT D: E: F:
```

If you wish to lock the drive D: until a user presses a key, specify:

```
UNMOUNT D: /k
```

If you wish to lock D and E, but unlock them with REMOUNT at separate times, specify:


```
UNMOUNT D: /nDDRIVE
UNMOUNT E: /nEDRIVE
```

If you wish to create a batch file that unmounts the C: drive, checks the return code and returns, it might look like:

```
UNMOUNT C: /p
if ERRORLEVEL 1 goto ERROR
Echo Successful lock - can begin SOS backup
goto SOS
:ERROR
Echo Unsuccessful lock - perform the SOS backup later
goto SOSLATER
...
```

C.3 REMOUNT.EXE

Run this program when you have run UNMOUNT and wish UNMOUNT to unlock the drive and refresh the operating system's view of it. It must be run in a separate MS-DOS prompt session as UNMOUNT.

Its form is:

```
REMOUNT [/nname]
```

Where:

- **/n<name>** is an optional parameter and is the same name specified when UNMOUNT was run.

In the example above, where UNMOUNT was run to lock D: E: F:, to unlock the drive, run:

```
REMOUNT
```

If the example above where you wish to unmount the drives separately run:

```
REMOUNT /nDDRIVE  
REMOUNT /nEDRIVE
```

Both UNMOUNT and REMOUNT return 1 if they fail and 0 if they succeed.

C.4 GENDATA.EXE

GENDATA is a bound program (DOS and OS/2) which allows you to quickly create a very large file (up to 4GB) and verify it's integrity. It's form is:

```
GENDATA <file name> <size in bytes> [/f[fill value]] [/r]
```

Where:

- **<file name>** is the fully qualified name of the file you wish to create or verify.
- **<size in bytes>** is the size of the file that you wish to create or verify.
- **/f[fill value]** is an optional parameter that allows you to be able to specify a single character to repeatedly place in the file. The fill value can be any number from 0 to 255. If you do not specify the /f option, then GENDATA will create the file with 4 byte numbers indicating the long word position. Specifying the /f option is faster than omitting it.
- **/r** causes GENDATA to verify the expected contents of a file. This option is generally used after a restore to verify that the data was replaced correctly.

For example, to create a 20,000,000 byte file named C:\TEST.DAT, specify:

```
GENDATA C:\TEST.DAT 20000000
```

To create a 1,000,000,000 byte file quickly named D:\TEST2.DAT, specify:

```
GENDATA D:\TEST2.DAT 1000000000 /f0
```

...and to verify the same file was restored correctly:

```
GENDATA D:\TEST2.DAT 1000000000 /f0 /r
```

D. FDRSOS - AIX Procedures

To use FDRSOS to back up or restore AIX attached EMC drives there are recommended procedures which help to guarantee that the data is complete and correct and the the operating system is properly notified.

Recommended procedures are provided for:

- Backup.
- Restoring disks to their original locations.
- Restoring one or more physical disks to different locations and scratching the original disks.
- Restoring a single volume volume group as a copy on the same system.
- Restoring a multiple volume volume group as a copy on the same system.

Note that coordination for the sequencing of host and workstation/server automated procedures is provided by FDR/UPSTREAM. FDR/UPSTREAM's USTBATCH host facility allows you to integrate workstation/server batch jobs into your host scheduling facilities. And FDR/UPSTREAM's host job execution facility allows host batch jobs to be executed from workstation/server automation. This allows you to coordinate the workstation/server procedures below with the execution of FDRSOS on the host. See chapter 8 (USTBATCH) in the FDR/UPSTREAM MVS manual and chapter 23 (Advanced FDR/UPSTREAM) in the FDR/UPSTREAM Workstation/Server manual for more information.

D.0.1 Recommended Backup Procedure

- ☐ Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode. You can use smit **umountfs** or the **umount** command to unmount file systems.

If the file systems can not be dismounted, you should insure that all cached I/O has been flushed out to the disk. Also, you should insure that no one is writing to the file system during the backup.

- ☐ If the physical disk(s) are in a volume group, deactivate the volume group. You can use smit **varyoffvg** or the **vary-offvg** command to deactivate a volume group. Again, if you can not deactivate the volume group, flush all I/Os to disk and verify that no users are writing to the file system.
- ☐ Run the SOS backups.
- ☐ Activate the volume group if the physical disk(s) are in a volume group. You can use smit **varyonvg** or the **vary-onvg** command to activate a volume group.
- ☐ Remount the file systems. You can use smit **mountfs**, smit **mountg** (if the file systems have been assigned to a group), or the **mount** command to mount a file system.
- ☐ Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

D.0.2 Restoring one or more physical disks to original locations

- ☐ Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode that are to be used for the restore. You can use `smit umountfs` or the `umount` command to unmount file systems.
- ☐ If the physical disk(s) are in a volume group, deactivate the volume group. You can use `smit varyoffvg` or the `varyoffvg` command to deactivate a volume group.
- ☐ Run the SOS restore. If you are restoring a multiple volume volume group, you MUST restore all of the volumes in the volume group.
- ☐ Activate the volume group if the physical disk(s) are in a volume group. You can use `smit varyonvg` or the `varyonvg` command to activate a volume group.
- ☐ Remount the file systems. You can use `smit mountfs`, `smit mountg` (if the file systems have been assigned to a group), or the `mount` command to mount a file system.
- ☐ Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

D.0.3 Restoring one or more physical disks to different locations and scratching the original disks.

- ☐ Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode. You can use `smit umountfs` or the `umount` command to unmount file systems.
- ☐ If the physical disk(s) are in a volume group, deactivate the volume group. You can use `smit varyoffvg` or the `varyoffvg` command to deactivate a volume group.
- ☐ If the physical disk(s) are in a volume group, export the volume group. You can use `smit exportvg` or the `exportvg` command to export a volume group.
- ☐ Clear the physical volume id's (PVID's) from both the original and target physical disk(s) BEFORE running the restore. This is VERY important. AIX will not tolerate duplicate physical volume id's. To clear a PVID, use the following form of the `chdev` command:

```
chdev -l hdisk# -a pv=clear
```

hdisk# represents the name of the physical volume such as `hdisk5`.
- ☐ Run the SOS restore. If you are restoring a multiple volume volume group, you MUST restore all of the volumes in the volume group.
- ☐ If the physical volume(s) are in a volume group, you must assign the physical volume id's of the restored physical volume(s). Note that physical volume id's will be the physical volume id's of the original volumes. Use `smit chgdisk` for EACH target volume. Make sure you change *ASSIGN physical volume identifier* to **yes**. AIX will use the physical volume identifier found on the restored volume.
- ☐ If the physical volume(s) are in a volume group, import the volume group. You can use `smit importvg` or the `importvg` command to import a volume group.

- ☐ Remount the file systems. You can use smit **mountfs**, smit **mountg** (if the file systems have been assigned to a group), or the **mount** command to mount a file system.
- ☐ Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

D.0.4 Restoring a single volume volume group as a copy on the same system.

- ☐ Ensure that target volume has been assigned a physical volume id BEFORE running the restore. Use smit **chgdsk** to display the physical volume id of the target volume. If it already has one, then cancel out of this smit screen. If not, change *ASSIGN physical volume identifier* to **yes** and perform the change.
- ☐ Run the SOS restore using the VOLRESET=NO option. SOS will preserve the physical volume id. To allow you mount the file systems on the copy at the same time as the originals, SOS will change the mount points of all file systems during the restore by appending **_sos** to the original mount point.

For example, if the original volume group had a file system with a mount point of **/myfilesys**, SOS will change it to **/myfilesys_sos**. This change will be picked up by AIX when the volume group is imported.

- ☐ Import the volume group. You can use smit **importvg** or the **importvg** command to import a volume group. Note that you can not use the name of the original volume group. You must assign a new name.
- ☐ You can now mount the filesystems on the new volume group. Remember that the mount points will be have **_sos** appended to the original mount point.

D.0.5 Restoring a multiple volume volume group as a copy on the same system.

- ☐ Clear the physical volume id's (PVID's) from target physical disk(s) BEFORE running the restore. To clear a PVID, use the following form of the chdev command:

```
chdev -l hdisk# -a pv=clear
```

hdisk# represents the name of the physical volume such as **hdisk5**.
- ☐ Run the SOS restore using the VOLRESET=NO option. You MUST restore all of the volumes in the volume group. SOS will assign new physical volume id's to all of the volumes in the volume group. To allow you to mount the file systems on the copy at the same time as the originals, SOS will change the mount points of all file systems during the restore by appending **_sos** to the original mount point.

For example, if the original volume group had a file system with a mount point of **/myfilesys**, SOS will change it to **/myfilesys_sos**.

This change will be picked up by AIX when the volume group is imported.

- ☐ You must assign the physical volume id's of the restored physical volume(s). Note that physical volume id's will be the physical volume id's set by SOS. Use smit **chgdsk** for EACH target volume. Make sure you change *ASSIGN physical volume identifier* to **yes**. AIX will use the physical volume identifier found on the restored volumes.
- ☐ Import the volume group. You can use smit **importvg** or the **importvg** command to import a volume group. Note that you can not use the name of the original volume group. You must assign a new name.

- ☐ You can now mount the filesystems on the new volume group. Remember that the mount points will be have _sos appended to the original mount point.